Ollscoil na Gaillimhe
University of Galway

Doctoral Thesis

Integrating Structured and Unstructured Knowledge Sources for Domain-Specific Chatbots

Rajdeep Sarkar

October 10, 2023

External Examiner	Supervisor	Supervisor	Internal Examiner
Dr. Ondrej Dusek	Dr. John McCrae	Dr. Mihael Arcan	Dr. John Breslin

Data Science Institute
College of Engineering and Informatics, University of Galway

"Take up an idea, devote yourself, struggle in patience, and the sun will rise."

Swami Vivekananda

ACKNOWLEDGEMENTS

The last 3 years of my life at the Data Science Institute, University of Galway have been the best years of my life. This period challenged me to perform my best both intellectually and physically.

I would like to begin by expressing my deepest gratitude to my esteemed supervisors, Dr John McCrae and Dr Mihael Arcan, for their unwavering support and guidance throughout my entire PhD journey. Their consistent encouragement, constructive feedback, and unwavering enthusiasm for my research were instrumental in providing me with a clear research direction leading up to the completion of this thesis. I would also thank Dr Paul Buitelaar, Dr Matthias Nickles and Dr Umair Ul Hassan for their constant support during my PhD.

I would like to express my sincere gratitude to Dr Sourav Dutta and Dr Haytham Assem for their guidance and support during my internship at Huawei Ireland Research Centre. Working with them was an invaluable experience that taught me the importance of being productive and adapting to new challenges quickly. I would also like to thank my colleagues and friends Koustava, Tarek, and Bernardo for engaging in thought-provoking discussions about life and research. Koustava, thanks for entertaining and discussing my research ideas. Special thanks to Tarek for being my gym buddy and to Bernardo for introducing me to the benefits of disciplined fitness. I am thankful to Sweta, Mashal, Gaurav, Priya, Rishab, Dhairya and Huan for the hilarious lunch breaks, river walks and night outs. I express my heartfelt gratitude to Sapna and Kartik for their constant support and guidance in making my stay in Ireland an enjoyable and enriching experience. I am especially grateful to Kartik for co-founding MustPool with me, which resulted in exponential growth and learning. Megha, thanks to your unwavering support and patient listening have helped me overcome many challenges. Piyush, thank you for teaching me the art of cooking delicious food and making my life easier with your culinary skills. Harshita, your friendship, and constant support during both good and bad times have helped me become a better individual, both professionally and personally.

I express my deep gratitude towards my mother, Anita, and father, Bishwajeet, for their unwavering support and encouragement throughout my academic journey. Their emphasis on the significance of education from my early years has shaped me into the person I am today. I am also indebted to my sister, Debosmita, whose wisdom and care for my family have been invaluable, particularly during the busy and demanding period of my PhD. Additionally, I extend my heartfelt thanks to my grandmother, Bula, and my late grandfather, Ratan, for instilling in me a love for mathematics from a young age. Their influence has been pivotal in my pursuit of academic excellence.

To Maa, Baba, Sana, Amma and Dadu

ABSTRACT

The increasing demand for customer support in various industries and the popularity of conversational interfaces has necessitated the development of chatbots. The availability of domain knowledge has enabled chatbots to understand and reason over complex concepts and contexts of specific domains enabling richer communication and engagement with customers. The integration of external knowledge is especially important in customer service, education and healthcare where accurate information is vital.

Knowledge-grounded chatbots can source external knowledge from either unstructured sources such as Wikipedia articles or structured sources like knowledge graphs. A significant hurdle is to effectively integrate such knowledge sources into the chatbots for enhanced user experience. Structured knowledge sources such as knowledge graphs are rich sources of information. However, an issue that arises during the integration of structured knowledge in chatbots is the need for careful consideration in constructing domain-specific subgraphs and selecting relevant knowledge. In contrast to structured data, utilising unstructured sources for knowledge-grounded chatbots presents different challenges, as it requires annotated data that is grounded in the domain, which necessitates domain expertise. It is crucial to develop systems that can adapt to available knowledge without the need for expert annotated datasets. Therefore, careful consideration must be given when selecting external knowledge sources for knowledge-grounded chatbots. Additionally, chatbots should be explainable, and their behaviour should be understandable to users. Furthermore, chatbots should leverage these knowledge sources effectively for informative and fluent response generation. This dissertation aims to present effective solutions for addressing the aforementioned problems during the integration of external knowledge into chatbots.

This dissertation proposes frameworks to effectively integrate external structured knowledge as well as unstructured knowledge into chatbots. We study the impact of by constructing contextually relevant subgraphs from a knowledge graph. Additionally, it presents frameworks to fuse unstructured knowledge into chatbots for question-answering without requiring manual annotation of datasets. Furthermore, it suggests using path traversal on a knowledge graph conditioned on chat context semantics for explainability, as the paths can convey context changes in a chat. Finally, a response generation methodology is proposed for generating informative, fluent, and coherent responses for knowledgeable response generation.

The integration of structured and unstructured knowledge is vital for developing effective chatbots, especially in domain-specific scenarios. Careful consideration must be employed for selecting the knowledge sources to ensure that these chatbots make optimal utilisation of the knowledge sources. Additionally, these systems must be explainable so that the behaviour is understandable to its users. The frameworks proposed in this thesis offer effective solutions for integrating structured and unstructured knowledge into chatbots while addressing the issues around explainability. The proposed response generation methodology also demonstrates the capacity of generating knowledge-grounded responses.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

DECLARATION

I declare that this thesis, titled "*Integrating Structured and Unstructured Knowledge Sources for Domain-Specific Chatbots*", is composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Galway, October 10, 2023

Rajdeep Sarkar

1 INTRODUCTION

Language is one of the basic building blocks of communication between human beings. Advancements in technological fields, especially in Natural Language Processing (NLP), have enabled humans to communicate with computers using their voice or through texts seamlessly. Personal assistants such as Siri, Alexa and Google Assistant have become commonplace with recent breakthroughs in NLP. These assistants facilitate conversations between users and the computer, either for answering queries (about the weather, product information etc) or for casual chatting purposes.

The creation of effective chatbots requires effective implementation of different NLP techniques [1] working in cohesion. This involves the chatbot's ability to comprehend the semantics and syntax of human language, possess an understanding of real-world knowledge, and respond to users with fluent and informative answers. With the advance of deep learning in NLP, various components of chatbots have moved towards neural network-based architectures [18].

In the field of modern-day chatbots, external knowledge is often incorporated to improve response generation[38]. These chatbots that utilise structured or unstructured external information are referred to as knowledge-grounded chatbots. The architecture of these chatbots includes two key components [73]: the *knowledge selector* and the *response generator*. The *knowledge selector* component is tasked with retrieving knowledge relevant to the dialogue context. It employs external knowledge from structured sources, such as knowledge graphs, or unstructured sources, such as documents or Wikipedia articles. After a comprehensive understanding of the conversation history and the selection of relevant knowledge, the *response generator* generates a response to the user's query. The effectiveness of each component is crucial to the success of the knowledge-grounded chatbot. This thesis investigates state-of-the-art models, prior research and proposes new techniques to address the challenges encountered in building knowledge-grounded chatbots.

1.1 MOTIVATION

Knowledge-grounded chatbots focus on leveraging external knowledge to generate appropriate, knowledgeable and informative responses to users. For exposition, considering the example in Figure 1.1, at turn 4, the chatbot needs to understand the user's information-seeking behaviour and guide the *knowledge selector* to traverse the KG and select relevant facts about the entity "*Pepe*". It needs to understand the meaning of the word "*teammate*" to understand that the athlete in discussion is "*Pepe*" and not "*Cristiano Ronaldo*". Additionally, the chatbot should be able to retrieve and utilise the fact that "*Pepe* scored 2 goals in 2015." to generate the final utterance in the dialogue. Hence, the *knowledge selector* and *response generator* should work in unison to respond with "He scored 2 goals in 2015." in accordance with the dialogue history.

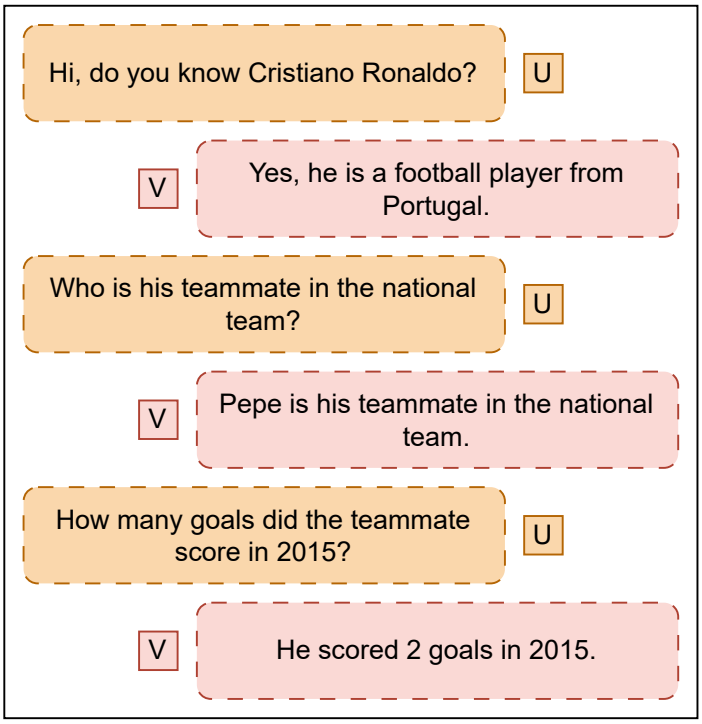

Figure 1.1: An example chat wherein a user is having conversation in the *sports* domain.

While structured knowledge bases are rich sources of information, they are not easy to obtain and can be expensive to build [50]. As a result of this, unstructured knowledge such as documents or Wikipedia articles can be utilised by the chatbot for informative response generation [23, 151] Figure 1.1 illustrates an instance wherein the chatbot has to comprehend the user requirements using the dialogue history. When posed with the user query *"Who is his teammate in the national team?"*, the chatbot needs to respond with appropriate facts in the response. When working in a knowledge-grounded setting, the chatbot must understand the semantic relationships between tokens in the chat history and the knowledge elements. The *knowledge selector* selects appropriate knowledge from external sources to guide the *response generator* for responding to the user. In this scenario, the chatbot needs to have an understanding of the dependency between the chat history and the external knowledge sources, select appropriate knowledge and then respond informatively to users. Additionally, it is beneficial to have an explicability module for providing seamless explanations of the responses generated by the chatbot as it can help to identify and correct errors or biases in the chatbot's decision-making process.

1.2 CHALLENGES

In this section, we outline several challenges associated with developing knowledge-grounded chatbots. A knowledge-grounded chatbot first selects appropriate knowledge from the available knowledge sources using the *knowledge selector* conditioned on the chat context. Thereafter, it utilises the retrieved knowledge and the chat context for generating appropriate responses leveraging the *response generator*. We investigated the impact of incorporating structured and unstructured knowledge on various elements of a chatbot. Our examination begins by analysing the

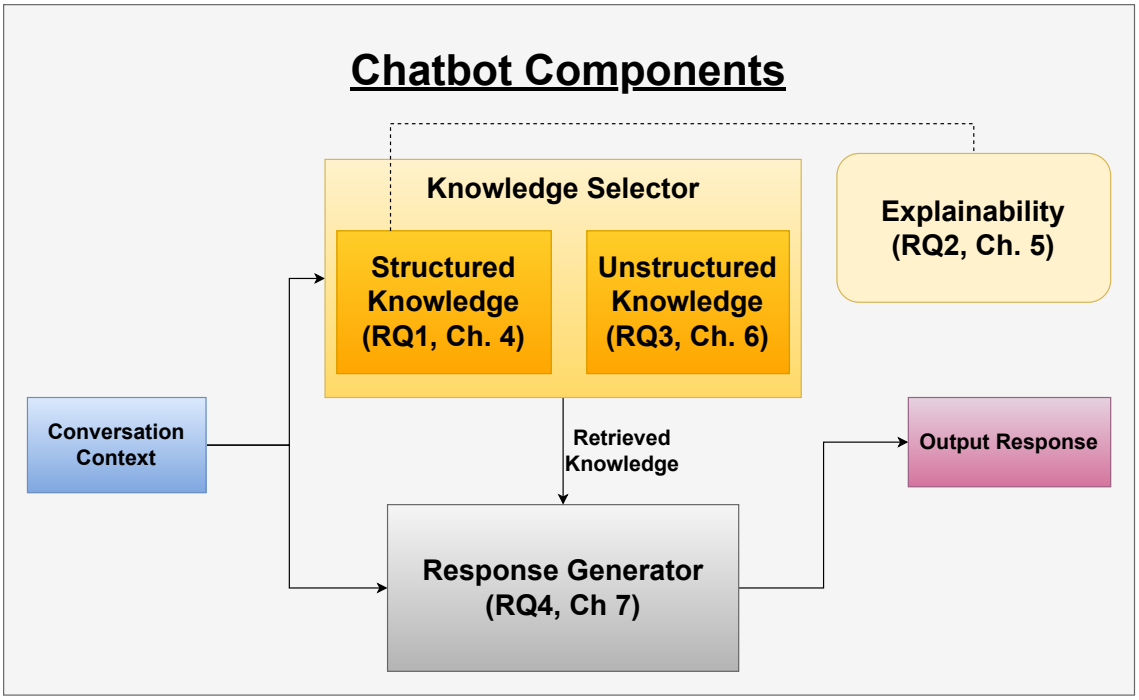

Figure 1.2: Outline of the research questions (RQs) pertaining to various components of a knowledge-grounded chatbot that have been addressed in this thesis.

obstacles associated with the size of contextual subgraphs of KGs when employing the *knowledge selector* component for selecting from structured knowledge sources. Afterwards, we scrutinise the hurdles of limited training data for domain-specific chatbots in the context of QA systems when utilising unstructured knowledge sources. Thereafter, we examine the barriers to the interpretability of responses generated by a knowledge-grounded chatbot. Finally, we identify the issues with developing domain-specific *response generators* through the integration of structured and unstructured knowledge sources.

1.2.1 Knowledge Selection from Structured Knowledge Sources

KGs are a rich source of structured information [42] which a chatbot can leverage to generate informative responses [170, 172, 141]. They contain information in the form of triples (h, r, t) where h and t are the head, and tail entities respectively, while r defines the relationship between them [42]. It is not uncommon for KGs to contain thousands of entities and millions of triples [42]. This leads to the chatbot searching for a relevant fact amidst millions of entries in the KG. However, a majority of information in the KG may not be relevant to the dialogue context [18, 174]. This challenge is more pronounced when building a domain-specific chatbot [18]. For example, when building a chatbot for the *sports* domain, information or facts about *Taj Mahal*[1] are rather irrelevant to the chatbot. Hence, it is essential to analyse the size and vocabulary of the KG and construct contextually relevant subgraphs. Additionally, the *knowledge selector* component would benefit from prioritising entities that have been mentioned more recently in the dialogue history as shown in Chapter 4. As an illustration, in Figure 1.1, the entity "*Pepe*" should be assigned higher importance than "*Cristiano Ronaldo*". The size of the KGs, and

1 https://en.wikipedia.org/wiki/Taj_Mahal

the presence of multiple entities in the context provide considerable barriers while building an efficient *knowledge selector*.

1.2.2 Explicability of Chatbot Responses

While present chatbot systems are effective at generating replies [171, 170], their internal decision-making can be difficult to understand, particularly for non-experts [46]. This lack of transparency can make it impossible to comprehend how a specific response was generated and limits the capacity to provide appropriate feedback [84]. Furthermore, as the complexity of these models grows, it becomes increasingly difficult to identify and rectify the flaws or biases in the responses. As a result, a focus on the explainability and transparency of chatbots is fundamental for a rich user experience.

KGs are a rich source of information as they convey semantic relationships between entities. Walk paths over a KG uncover semantic relationships between entities that might not be apparent from a simple inspection of the graph [84]. The walk paths are paths over a KG that originate from entities or facts mentioned in the dialogue history and terminate at entities to be mentioned in the response text. These paths can be utilised to provide an explanation of the flow of the conversation with the chatbot. Procedures for providing such explanations either do not include the KG structure into account or are computationally expensive to train and evaluate as illustrated in Chapter 5. Therefore it is necessary to develop reliable architectures for generating KG paths for dialogue explanation.

1.2.3 Knowledge Selection from Unstructured Knowledge Sources

The utilisation of KGs can serve as a valuable structured source of information to generate natural and coherent responses [171, 84, 141, 172], thereby playing an important role in developing chatbots. However, the creation and maintenance of KGs are resource intensive and expensive as they often require specialised knowledge and expertise. There is thus a substantial expense in terms of iteration time to incorporate new information and specialised staff associated with this process [50, 78].

Responding to users' queries is an important use-case of chatbots. The QA component of a chatbot provides relevant information to the users' queries. The QA component may utilise external knowledge to respond to users' queries. In the context of knowledge-grounded chatbots, there exist primarily two types of QA modules: Extractive Question Answering (EQA) module and Answer Sentence Selection (AS2) module. EQA refers to the task of retrieving the exact answer to a user query from given contexts or documents, while AS2 refers to selecting a sentence or sentences from documents that can potentially answer a user's query. In the case of either EQA or AS2, the system chooses the text span or sentences from a document or an article that can potentially respond to the user's query appropriately [157, 58]. QA systems trained on generic open-domain datasets fail to be effective in a domain-specific setting due to the presence of domain-specific terminologies as shown in Chapter 6. On the other hand, developing domain-specific QA modules, and constructing domain-specific structured knowledge can be expensive and time-consuming to develop [78]. The annotation of QA datasets by domain experts can pose considerable hurdles in the context of domain-specific knowledge, resulting in extended data

curation time and the necessity for domain specialists [138]. Thus there is a need to develop frameworks for domain-adaptation of QA components without the need for manually labelled training datasets.

1.2.4 Response Generation using Structured and Unstructured Knowledge

As highlighted in the preceding sections, knowledge-grounded chatbots revolve around the utilisation of structured or unstructured knowledge for generating responses. This process includes the *knowledge selector* component that retrieves relevant knowledge based on the dialogue context and the *response generator* component that utilises the retrieved knowledge and the dialogue context for producing coherent and fluent responses. One of the primary challenges in this area is the integration of knowledge from diverse sources [172, 141], including KGs and documents. Moreover, building language models for domain-specific applications necessitates fine-tuning on datasets from that specific domain as shown in Chapter 6. This requirement creates a significant bottleneck to scalability since fine-tuning language models with millions of parameters is resource-intensive [61]. In addition, developing methodologies for multiple domains requires creating, managing, and storing individual models for each domain, which can be costly in terms of both disk space and training duration. Therefore, it is crucial to devise parameter-efficient learning techniques that can generate informative and coherent responses by leveraging knowledge grounding.

1.3 CONTRIBUTIONS

In this thesis, we aim to tackle the challenges discussed in Section 1.2 that arise in the domain of knowledge-grounded chatbots, specifically with regard to knowledge selection and response generation. To achieve this goal, we have formulated a set of research questions and outlined our proposed methodologies as follows:

- **RQ1**: Can we leverage subgraph selection from a KG to improve the performance of knowledge selection for chatbots by identifying the most contextually relevant subgraph?
- **RQ2**: Can we utilise walk paths over a KG for generating an explanation to a chatbot conversation using a computationally efficient framework?
- **RQ3**: Can we leverage unstructured knowledge sources such as documents or paragraphs to improve the performance of QA components of chatbots without the need for a manually annotated domain-specific training dataset?
- **RQ4**: Can we utilise context modelling and parameter-efficient tuning of language models while leveraging structured and unstructured knowledge for knowledgeable response generation in chatbots?

1.3.1 RQ1: Knowledge Selection from Structured Sources

We investigate the impact of the KG size and vocabulary on the retrieval performance of the *knowledge selector* in a knowledge-grounded chatbot, with a focus on the domain of movie recom-

mendation in conversational settings. We present a subgraph construction approach to enrich information about movie entities, which achieves similar performance to using the entire knowledge graph. This is due to the removal of extraneous noise from the graph. Additionally, we propose a recency-based model for movie recommendation that considers both the dialogue context and the recency of entities mentioned in the dialogue. Our research's efficacy is demonstrated by conducting comprehensive experiments on a dataset for conversational recommendation grounded in a KG.

Publications:

1. Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. 2020. "Suggest me a movie for tonight: Leveraging knowledge graphs for a conversational recommendation". In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4179–4189.

1.3.2 RQ2: Explicability of Chatbot Responses

Addressing the issues detailed in Section 1.2.2 with regards to the explainability of chatbot responses, we propose *KG-CRuSE*, a lightweight yet effective framework for generating explanations to conversations utilising walk paths over a KG. Our proposed approach uses an LSTM network 2.4.4 for traversing the KG by leveraging semantically enriched embeddings of the dialogue history and the KG elements. *KG-CRuSE* has a small number of model parameters, and its contextual semantic embeddings lead to improved performance on the dialogue explicability task. Additionally, the framework has lower training and inference computing requirements as compared to other competing methodologies.

Publications:

1. Rajdeep Sarkar, Mihael Arcan, and John P. McCrae. "KG-CRuSE: Recurrent Walks over Knowledge Graph for Explainable Conversation Reasoning using Semantic Embeddings". In: *Proceedings of the 4th Workshop on NLP for Conversational AI, ConvAI@ACL 2022*. Association for Computational Linguistics, 2022, pages 98–107.

1.3.3 RQ3: Knowledge Selection from Unstructured Sources

For the second research question, we focus on the utilisation of unstructured knowledge in the QA modules of a knowledge-grounded chatbot. We propose *Qasar*, a self-supervised EQA framework employing synthetically generated question-answer pairs from domain-specific documents for domain adaptation, negating the need for manually annotated question-answer pairs. Furthermore, the proposed framework reduces the noise in the knowledge source by eliminating sentences with low semantic similarity in regard to the query. We also propose *SEDAN*, a self-supervised AS2 framework that utilises synthetic question-answer sentence pairs for domain adaptation. *SEDAN* finetunes a sentence transformer for selecting answer sentences using a synthetically generated dataset during domain adaptation of the AS2 module. The effectiveness of the *Qasar* and *SEDAN* frameworks in EQA and AS2 tasks, respectively, is demonstrated through extensive experiments on open- and closed-domain datasets.

Publications:

1. Haytham Assem*, Rajdeep Sarkar*[2], and Sourav Dutta. 2021. "Qasar: Self-Supervised Learning Framework for Extractive Question Answering". In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, pages 1797–1808.

2. Rajdeep Sarkar, Sourav Dutta, Haytham Assem, Mihael Arcan, and John P. McCrae. "Semantic Aware Answer Sentence Selection Using Self-Learning Based Domain Adaptation". In:*KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. ACM, 2022, pages 3849–3857.

1.3.4 RQ4: Response Generation using Structured and Unstructured Knowledge Sources.

We propose a new framework, called *PICKD*, to generate responses that address the challenges discussed in Section 1.2.4 by utilising both structured and unstructured knowledge sources. The framework comprises two components, the *Knowledge Selector* and the *Response Generator*. The *Knowledge Selector* employs an innovative *In-Context* prompt tuning paradigm that enables it to select relevant knowledge from both structured and unstructured sources without fine-tuning the language model parameters used for retrieval. The response generator leverages the selected knowledge and the conversation history to generate an appropriate response. The integration of structured and unstructured knowledge sources in *PICKD*, while leveraging the parameter-efficient tuning process enables the generation of knowledgeable and coherent responses.

Publications:

1. Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. "PICKD: In-Situ Prompt Tuning for Knowledge-Grounded Dialogue Generation". In: *PAKDD '23: The 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2023, pages 124-136.

1.4 OUTLINE

The remainder of the thesis is organised as follows. Chapter 2 provides foundational knowledge on chatbots, structured and unstructured knowledge sources, and deep learning techniques relevant to this thesis. Chapter 3 introduces the state-of-the-art methodologies developed for *knowledge selector*, *response generator* and knowledge-grounded chatbots. Chapter 4 discusses our study on the knowledge selection for chatbots, in the realm of Conversational Recommender Systems (CRSs), using structured knowledge sources. Chapter 5 puts forth a novel and efficient framework for response explainability of chatbots using KGs. Chapter 6 presents effective methodologies for domain adaption of QA components of a chatbot using self-supervised learning. Chapter 8 exhibits our innovative framework for generating responses utilising both structured and unstructured sources of knowledge. Finally, Chapter 9 provides a summary of our contributions and proposes possible directions for future work.

2 * Haytham Assem was responsible for planning the project, conducting the initial experiments and writing the paper. Rajdeep Sarkar was responsible for designing and conducting the experiments and writing the paper.

1.5 OTHER PUBLICATIONS

Other publications that were published during this thesis duration, but not directly relevant to the thesis are enumerated below:

1. Rajdeep Sarkar, Atul Kr Ojha, Jay Megaro, John Mariano, Vall Herard, and John Philip McCrae. "Few-shot and Zero-shot Approaches to Legal Text Classification: A Case Study in the Financial Sector". In: *Proceedings of the Natural Legal Language Processing Workshop 2021*. 2021, pages 102–106.

2. Tapan Auti, Rajdeep Sarkar, Bernardo Stearns, Atul Kr Ojha, Arindam Paul, Michaela Comerford, Jay Megaro, John Mariano, Vall Herard, and John Philip McCrae. "Towards Classification of Legal Pharmaceutical Text using GAN-BERT". In: *Proceedings of the First Computing Social Responsibility Workshop within the 13th Language Resources and Evaluation Conference*. 2022, pages 52–57.

3. John P. McCrae, Pranab Mohanty, Siddharth Narayanan, Bianca Pereira, Paul Buitelaar, Saurav Karmakar, and Rajdeep Sarkar. "Conversation Concepts: Understanding Topics and Building Taxonomies for Financial Services". In: *Inf.* 12.4 (2021), pages 160.

4. Koustava Goswami, Rajdeep Sarkar, Bharathi Raja Chakravarthi, Theodorus Fransen, and John P. McCrae. "Unsupervised Deep Language and Dialect Identification for Short Texts". In: *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*. International Committee on Computational Linguistics, 2020, pages 1606–1617.

2 BACKGROUND

This chapter provides an overview of the basic tools necessary for understanding the thesis. In Section 2.1, we begin with analysing different kinds of chatbots in use in recent times. We then discuss KGs in Section 2.2. Thereafter in Section 2.3, we study the basic concepts required for understanding the basic workings of a neural network. Thereafter, in Section 2.4, we study the different text encoders utilised for capturing contextual information from texts built using neural networks. In Section 2.5, we study a few large language models that have been shown to be effective in modelling textual information. Finally, Section 2.5.6 introduces neural network architectures that have been utilised for response generation.

2.1 CHATBOTS

Chatbots are tools enabling effortless interactions with computers. Chatbots utilise NLP and machine learning methodologies to retort to user requirements using human-like responses [2, 89]. They are often used as virtual assistants, customer service agents, or language learning tools, among other applications. In the realm of chatbots, the primary objectives are to engage in casual conversations with users, assist users in accomplishing specific tasks by comprehending their requirements, or engage in topic-based conversations while drawing from external sources of knowledge [2]. We discuss these different genres of chatbots in the following sections.

2.1.1 Chit-chat Chatbots

Chit-chat conversations are a type of communication wherein the participants engage in small talk or casual conversation [147, 148]. These chats involve exchanging pleasantries, exchanging information about a particular topic or making general observations or comments, without having a specific agenda or a goal in place that needs to be achieved [147]. Chit-chat chatbots engage in discourse with users without the goal of addressing a specific task.

In Figure 2.1a, a chit-chat conversation between a chatbot and a user is depicted. The conversation involves the user initiating the chat, exchanging pleasantries with the chatbot, and subsequently requesting information about the chatbot's movie and pet preferences as well as weather information.

To develop a chit-chat chatbot, it is imperative to gather a comprehensive corpus of conversational data that encompasses various personalities, styles, emotions, and roles. It is crucial to ensure that the dataset is diverse enough to facilitate the generation of distinct response styles by the model. The collected corpus can then be used to train a chatbot model that can effectively comprehend the subtleties of human conversation styles and provide relevant responses.

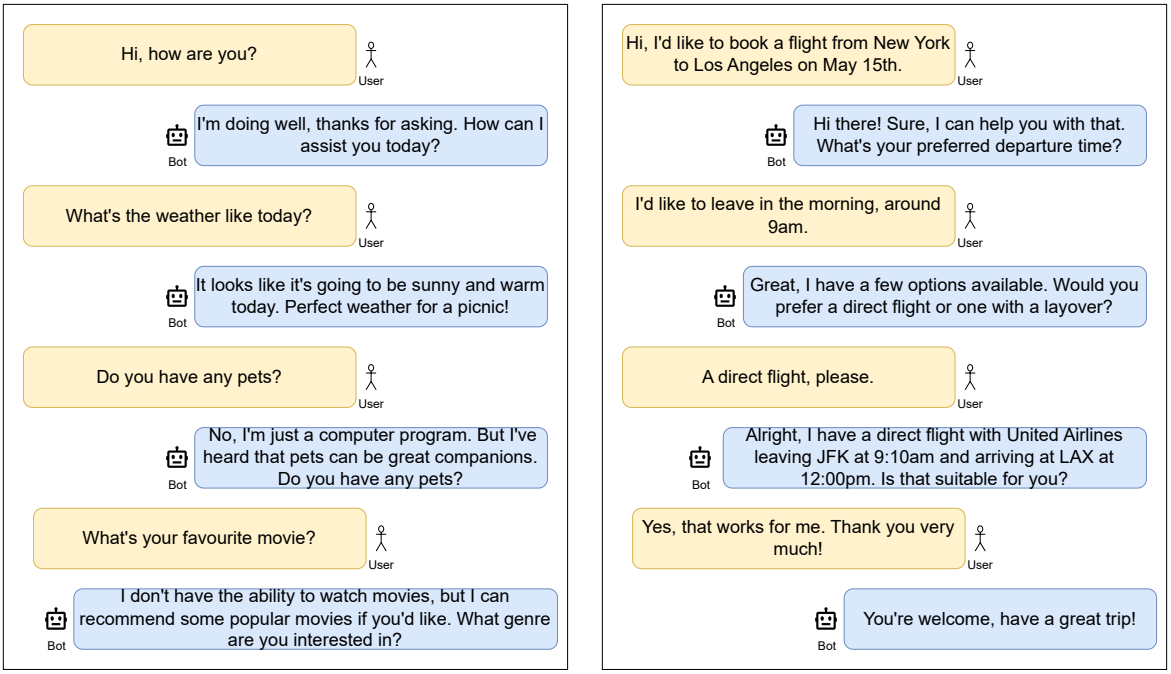

Figure 2.1: Example of different user conversations with a chatbot. (a) A scenario wherein the chatbot carriers out a chit-chat conversation with the user. (b) A case where the chatbot is tasked with booking flight tickets for users.

2.1.2 Task Oriented Chatbots

Task-oriented conversations between a user and a system have a specific objective, such as booking a hotel, scheduling an appointment or reserving a flight [137]. Task-oriented chatbots guide users through a series of structured steps to help them achieve their goal efficiently. In comparison to open-domain conversations or chit-chat conversations, task-oriented conversations have a more structured conversation flow [149].

As depicted in Figure 2.1b, a task-oriented chatbot can facilitate flight booking by prompting users to provide specific details, such as their destination, preferred travel dates, and flight type. The chatbot then uses its domain knowledge to suggest flight options that align with the user preferences and completes the booking process. Designing effective task-oriented chatbots requires a deep understanding of the relevant domain, as well as the potential user requirements and preferences.

Task-oriented chatbots leverage natural language processing and machine learning techniques to accurately interpret user input and provide appropriate responses. The development of task-oriented chatbots involves constructing a domain-specific knowledge base [149, 137], which includes information about the task or domain the chatbot is designed to assist with. Moreover, the chatbot needs to be trained on a wide range of possible user queries and responses to ensure it can accurately assist users with their desired task. Such chatbots can be integrated into various applications, such as messaging apps, websites, or mobile apps.

2.1.3 Knowledge Grounded Chatbots

Knowledge-grounded conversations utilise external knowledge sources for responding informatively in the conversation [18, 23, 170]. In contrast to chit-chat chatbots that generate responses solely using the conversation context, knowledge-grounded chatbots utilise external knowledge sources in addition to the conversation context for responding to the user [174]. This external knowledge can take the form of structured data such as KGs or unstructured text data such as Wikipedia articles for generating accurate and fluent responses [141, 172].

In Figure 2.2 (a), an example is presented wherein the conversation is grounded on an unstructured knowledge source. Typically, unstructured sources are in the form of text sequences extracted from Wikipedia articles, documents, or domain-specific articles. The chatbot processes the context of the chat, models context-dependent information from these knowledge sources for response generation. The chatbot in the example is provided with textual information about the *Eiffel Tower*. It processes information from different sentences in the text, comprehends the conversation context, and generates an appropriate, informative, and fluent response for seamless user interaction.

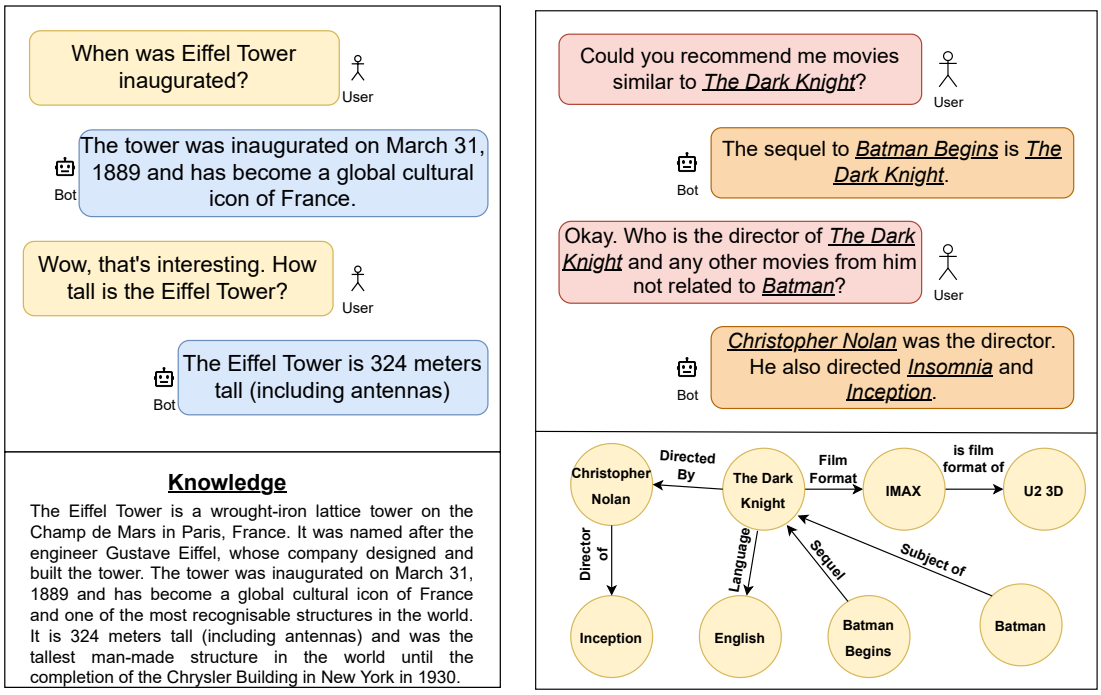

Figure 2.2: Example of knowledge-grounded conversations. (a) An example instance wherein the chatbot utilises unstructured knowledge for generating informative responses. (b) The chatbot utilises facts from a KG for generating knowledgeable responses.

Similarly, in Figure 2.2 (b), the chatbot uses structured knowledge from a Knowledge Graph (KG) to respond to user queries. The chatbot begins by linking the entities mentioned in the conversation to the KG, and then leverages the conversation context and KG information to generate responses that enhance user experience. KGs have been successfully utilised for answering queries [108] and recommendation [18, 174, 175] tasks among others.

2.2 KNOWLEDGE GRAPHS

KGs are a rich source of structured information, that aim to capture and model information in a form that is both machine-readable and human-understandable [125, 41]. A KG enables understanding of relationships between different entities and concepts. Typically KGs have a graph structure with multiple nodes and edges connecting them. Nodes represent different entities or concepts while the edges denote the relationship between two entities or concepts [41].

Typically, KGs are constructed using a combination of automated and manual methods. The procedure begins with the identification of relevant data sources, such as structured and unstructured text documents, databases, and existing KGs. The information is subsequently analyzed and structured using techniques such as named entity recognition, relation extraction, and entity disambiguation.

Once the data has been formatted, it can be integrated into the knowledge graph using methods such as entity alignment, which maps entities from several sources to a common identifier, and property mapping, which maps properties from many sources to a common vocabulary [42, 41]. After the knowledge graph is created, it can be refined and expanded via manual curation and expert input. Adding new entities and relationships, correcting discrepancies, and testing the graph's accuracy are all examples of this.

Figure 2.2b features a KG from the movie domain. We notice that the edge "*Christopher Nolan*, *Director of*, *Inception*" conveys the information that *Christopher Nolan directed the Inception movie*. This graph information can be effectively utilised by a computer to learn real-world information. KGs have been effectively used in question-answering, conversational agents, recommendation and retrieval tasks.

2.3 NEURAL NETWORKS

Neural networks are a machine learning technique very loosely inspired by the functioning of biological neurons [53]. They consist of nodes or neurons that process data through multiple layers of computation. Neural networks are capable of capturing complex feature relationships that may be difficult to detect through traditional methodologies and have achieved state-of-the-art performance in various fields such as NLP, computer vision, and speech processing. This section describes the fundamental concepts of feed forward networks, activation functions, loss functions, gradient descent and backpropagation that are necessary to understand the neural architectures employed in this thesis.

2.3.1 Feed Forward Networks

Feed Forward Networks (FFN) [12] are a class of neural networks used extensively for classification and regression tasks. Each layer consists of a set of nodes connected to the nodes of its preceding layer. Each layer receives an input, processes it through an activation layer which typically introduces non-linearity, and passes this output to the next layer. FFNs do not contain

loops within the network architecture and the output of a layer is used as an input to the next layer. Mathematically an FFN layer is defined as:

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input vector, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^m$ is the bias vector, $\mathbf{h} \in \mathbb{R}^m$ is the output vector, and f is a nonlinear activation function applied element-wise to the elements of $\mathbf{W}\mathbf{x} + \mathbf{b}$. During the training phase, we learn the parameters of $\mathbf{W}$ and $\mathbf{b}$ to fit the FFN on the dataset.

2.3.2 Activation Functions

Activation functions are mathematical functions which are applied to the output of a node in a neural network for introducing non-linearity in the network. Sigmoid, tanh and ReLU are some common examples of activation functions.

The sigmoid activation function [86] is a commonly used activation function with range (0, 1), and is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

ReLU [4, 85] is another commonly used activation function due to its simplicity and computational efficiency and is defined as:

$$f(x) = max(0, x) \tag{2.3}$$

Another less commonly used activation function is the hyperbolic tangent function or the tanh activation function having range (-1, 1) and is defined as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.4}$$

2.3.3 Loss Functions

Loss functions provide an estimation of the model's ability to accurately map inputs to outputs. In the process of optimising a neural network architecture, we learn parameters that minimise a given loss function. A lower value of the loss function generally indicates greater confidence in the model's ability to make accurate predictions for a given input.

Cross entropy loss is a commonly used loss function while optimising neural networks [124]. It measures the difference between the distribution of the predicted labels from the distribution of the true labels. In the case of binary classification, the *cross entropy loss* is defined as:

$$L = -y \log(p) - (1 - y) \log(1 - p) \tag{2.5}$$

where y is the true label (either 0 or 1), p is the predicted probability of the positive class (between 0 and 1) and log is the natural logarithm. The goal while training the model is to

reduce the value of the *cross entropy loss* between the predicted and true labels using gradient descent optimisation (Section 2.3.4).

Another commonly used loss function is the *mean squared error loss*. It measures the average squared difference between the predicted and actual values. The formula for mean squared error loss is:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.6}$$

where n is the number of training samples, y_i is the true value and $\hat{y}_i$ is the predicted value. We learn a model by minimising this loss using gradient descent optimisation Section 2.3.4.

2.3.4 Gradient Descent

Gradient descent [106] is an optimisation algorithm used for achieving the minimum value of a function. Given a function $f(x)$, gradient descent iteratively updates the value of x so that the minimum value of f is achieved. During every iteration, the update takes place in the direction of the negative gradient as it represents the direction of the steepest slope denoting the value of the function decreases the fastest in that direction. Mathematically it is defined as follows:

$$x_{t+1} = x_t - \alpha\frac{\partial f_{x_t}}{\partial x} \tag{2.7}$$

where α is the learning rate, x_t and x_{t+1} is the value of x at timestep t and $t+1$. During every iteration, gradient descent first computes the partial derivative of f with respect to x evaluated at x_t. This is then multiplied with the learning rate α to get the updated value of x. The learning rate α denotes the step size of the update.

The iterative procedure is carried out until either a predetermined number of iterations have been executed or the convergence criteria has been met. An illustration of a convergence criterion could be the scenario where the absolute difference between x_{t+1} and x_t is less than a specified small value ϵ (e.g., 1e-4 or 1e-5). The final value of x is assumed to be the local minima of f if it is a convex function.

The gradient descent algorithm works towards minimising the value of the function, but it has a few weaknesses. It does not remember the gradients of the previous step and has the same learning rate for all the parameters resulting in a longer convergence time. Kingma and Ba [51] posited the Adam optimiser, which utilises an adaptive learning rate for each parameter, thus resulting in faster and more robust optimisation.

2.3.5 Backpropagation

Neural networks utilise the backpropagation algorithm [81] for learning the learnable weights. Backpropagation is used to compute the gradients of the learnable weights with respect to the loss function, which is then used to learn the weights using optimisation algorithms such as the gradient descent algorithm.

The backpropagation algorithm computes the gradients of the loss function with respect to the weights at each layer of the network by utilising the chain rule of calculus. This rule states that the derivative of a composite function f made up of a sequence of functions $f_1, f_2, ..., f_n$

with respect to a variable x can be determined by multiplying the derivatives of each function f_i with respect to the output of the previous function f_{i-1}.

$$\frac{\partial f}{\partial x} = \frac{\partial f_n}{\partial f_{n-1}} \frac{\partial f_{n-1}}{\partial f_{n-2}} \cdots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x} \tag{2.8}$$

In the context of a feedforward neural network, the function f represents the output of the network given an input, and the functions $f_1, f_2, ..., f_n$ represent the computations performed by the layers of the network. The variables x represent the weights of the network.

2.4 NEURAL TEXT ENCODERS

Text encoders [65, 70] are a set of techniques that are employed to transform sentences or documents into a fixed-size vector representation, which is subsequently used as input to a machine learning model. The main objective of text encoders is to capture both the semantic and syntactic information of the sentence(s) in a numerical representation. This section introduces the readers to word embeddings, wordpiece embeddings, recurrent neural networks and the transformer architecture used for text encodings.

2.4.1 Word Embeddings

In NLP, words form the basic units of a sentence. Neural networks require an effective method to process words in sentences. To achieve this, each word w_i in a sentence is associated with a vector of fixed dimension d, known as the word embedding of w_i. Word embeddings [97, 80] are typically stored in a word embedding matrix $\mathbf{X} \in \mathbb{R}^{|V| \times d}$, where V represents the vocabulary and $|V|$ denotes the size of the vocabulary. A sequence of words w_1, w_2, ..., w_T is then represented as $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_T$, where $\mathbf{x}_i$ is the word embedding of w_i. This vector representation of the word sequence is then sent as input to the machine learning model.

2.4.2 Wordpiece Embeddings

While the word embeddings can capture the contextual information in a text sequence, they are not optimal for out-of-vocabulary words, or words occurring rarely in the contexts [142]. Wu et al. [142] proposed dividing the words into *sub-words* (*wordpieces*). This decomposition of words into small units or sub-words enables improved handling of out-of-vocabulary or rare words. For example, consider the sentence "I like sitting in a rickshaw". The word-piece representation of the sentence might look like ["I", "like", "sitting", "in", "a", "rick", "#shaw"]. Even though the word "rickshaw" might not have appeared in the vocabulary, models can still understand the context by composing it from the subword tokens.

2.4.3 Recurrent Neural Network

The feed-forward neural network operates under the assumption that input features are independent of one another, and uses them to learn a given task. However, in the domain of natural

language processing, words within a sentence exhibit semantic dependencies on one another, rendering them non-independent. Moreover, the variable length of text sequences poses challenges in using feed-forward networks since the number of parameters scales with the sequence length. Training models with long text sequences becomes difficult due to the large number of parameters involved. To capture this sequential information, Elman [27] introduced Recurrent Neural Networks (RNNs), which are designed to model sequences of data. RNNs leverage sequential information to learn a representation of the input sequence. RNN maintains a state representation $\mathbf{h}_t$ representing the sequential information captured at time-step t. Mathematically it is defined as:

$$\mathbf{h}_t = \sigma_x(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.9}$$

$$y_t = \sigma_y(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y) \tag{2.10}$$

where σ_x and σ_y are activation functions. $\mathbf{W}_h$, $\mathbf{U}_h$, $\mathbf{W}_y$ are the weights to be learned, while $\mathbf{b}_h$ and $\mathbf{b}_y$ are the bias terms to be learned. $\mathbf{x}_t$ is the input representation at time-step t and $\mathbf{y}_t$ is the output at time-step t. It is important to notice that RNN utilises input representation $\mathbf{x}_t$ of time t and the state representation $\mathbf{h}_{t-1}$ of time $t-1$ is utilised for computing $\mathbf{h}_t$ as showcased in Equation 2.9. This recurrent relation between adjacent time-steps allows for the capture of dependencies between time-steps.

2.4.4 Long-Short Term Memory

While RNNs have the capability to model sequential data, Long-Short Term Memory (LSTM) [40] are preferred as they are better at capturing long-range dependencies in sequential data. Additionally, LSTMs can be more robust to noisy data than RNNs, because the gating mechanisms allow them to selectively filter out irrelevant information and retain only the most important features. Mathematically the workings of an LSTM are defined as follows:

$$i_t = \sigma(\mathbf{W}_i[x_t, h_{t-1}] + \mathbf{b}_i) \tag{2.11}$$

$$f_t = \sigma(\mathbf{W}_f[x_t, h_{t-1}] + \mathbf{b}_f) \tag{2.12}$$

$$o_t = \sigma(\mathbf{W}_o[x_t, h_{t-1}] + \mathbf{b}_o) \tag{2.13}$$

$$g_t = \tanh(\mathbf{W}_g[x_t, h_{t-1}] + \mathbf{b}_g) \tag{2.14}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.15}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.16}$$

where σ and tanh are the sigmoid and the hyperbolic tangent activation function respectively, while $\odot$ is the element-wise product, commonly known as Hadamard product. The weights $\mathbf{W}_i$, $\mathbf{W}_f$ and $\mathbf{W}_o$ along with the bias terms $\mathbf{b}_i$, $\mathbf{b}_f$ and $\mathbf{b}_o$ are the parameters to be learned. In this case, i_t, f_t and o_t are known as the input gate, forget gate and output gate respectively, while g_t and c_t are known as the *new memory cell* and *final memory cell* respectively. The *new memory cell* utilises the *final memory cell* state of the previous step and the input at the current step to generate the *new memory cell*. This *new memory cell* captures the aspects of the new input along with the previous contexts. Gate i_t utilises the current input x_t and the h_{t-1} to decide if the current input is worth preserving and should be used to gate the *new memory*. Similarly, gate

f_t decides if the previous memory cell c_{t-1} should be utilised when generating the new cell state. The final memory cell c_t contains information, all of which might not be useful for the representing h_t, hence the output gate o_t makes an assessment of the amount of information from c_t is to be presented in h_t.

Two significant challenges with RNNs and LSTM networks are the vanishing gradients and exploding gradients problem. When gradients are propagated backwards through time, they can diminish or increase exponentially as they involve multiplication with gradients from the later steps, leading to information loss or saturation. This results in the information from earlier timesteps being unlearned, severely limiting the network's ability to capture long-term dependencies and perform well on downstream tasks. While LSTM networks were designed to address the vanishing gradient problem in RNNs, researchers have proposed methods such as the use of ReLu activation function in RNNs, which allows gradients to flow more easily through the network by avoiding saturation. Additionally, weight initialisation and the use of residual networks have also been explored as possible solutions to the vanishing gradient problem [120, 160]. Kanai et al. [48] suggested the use of gradient clipping, wherein whenever gradients reach a certain threshold, they are set back to a small number to address the gradient explosion problem in RNNs and LSTMs.

2.4.5 Transformers

A significant limitation of utilising RNNs or LSTMs for NLP tasks is that they process inputs sequentially making them slower than models that can process sequences parallelly. Additionally, they use recurrence for computing the states which are dependent on the previous timesteps. Such limitations led to research developing the transformers architecture [130].

Transformers are a class of neural architectures which utilise self-attention [94] for computing the sequence representation. The architectural design of transformers allows the processing of tokens in parallel instead of sequential computations. The self-attention mechanism enables transformers to capture long-range dependencies between tokens instead of relying on recurrent computations as used in LSTMs. Mathematically the internal workings of a transformer block are defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2.17}$$

where Q, K, and V are the query, key, and value learnable weight matrices respectively, and h is the number of attention heads. The attention heads learn independently of each other, enabling each head to focus on different aspects of the input text and capture distinct information. Each token is associated with query, key, and value embeddings. The multiplication of the query matrix Q and the transposed key matrix K^T captures the interrelatedness of each word with every other word in the sequence. This enables the model to assess the significance of other words in the context of a given word. The resulting matrix is scaled down (as shown in Equation 2.19) and passed through a softmax layer to obtain the attention weights. These attention weights are then used to compute the contextual representation of each token in the sequence by multiplying them with the value matrix.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.18}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.19}$$

where W_i^Q, W_i^K, and W_i^V are learnable weight matrices for the i-th attention head. The output of each attention head is concatenated along the feature dimension, and then linearly transformed by a learnable weight matrix W^O.

After the multi-head attention mechanism, the output is passed through a feedforward neural network (FFN) as in the standard transformer encoder block:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{2.20}$$

where x is the input to the MLP, W_1, b_1, W_2, and b_2 are the weight and bias parameters to be learned, and ReLU is the rectified linear unit activation function. Finally, the output of the MLP is added to the input using residual connections and layer normalization:

$$o_x = \text{LayerNorm}(x + \text{FFN}(\text{MultiHead}(Q, K, V))) \tag{2.21}$$

It is important to note that the above-described approach does not consider the positional information of tokens. To incorporate spatial information into the representation, positional embeddings are added to the input x. The transformer architecture proposed by Vaswani et al. [130] uses sinusoidal positional embeddings. However, other studies [136] have employed learnable positional embeddings for this purpose.

2.5 LANGUAGE MODELS

Language Models (LMs) [144] are models trained on large amounts of text data to capture the likelihood of a word given its context. These models are designed to capture the statistical and semantic relationships between different tokens in the text. Transformer-based LMs have proven to be powerful tools in modelling language and have been effectively applied in various NLP tasks such as machine translation, chatbots, text generation, and text classification. The stacking of multiple transformer layers increases the model's capacity and allows it to capture intricate dependencies in the data. LMs are trained using self-supervised learning, where they learn from a corpus without the need for explicit human-annotated data. In this section, we discuss the transformed-based LMs used in this thesis. Typically LMs are trained on a large corpus of text [22, 72, 99] and the equation governing the training is given as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log P(w_{i,t}|w_{i,<t};\theta) \tag{2.22}$$

where $\mathcal{L}$ is the loss function, N is the number of training examples, T_i is the length of the i-th example, $w_{i,t}$ is the t-th word in the i-th example, $w_{i,<t}$ is the sequence of words before the t-th word in the i-th example, and θ are the parameters of the language model. The goal of training is to minimise this loss function with respect to the parameters θ.

2.5.1 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) [22] is a transformer-based pre-trained LM that learns to represent a text by capturing the semantic relationships between its tokens. BERT is trained in an unsupervised manner on a large corpus of text, enabling it to learn rich representations of language. This makes it possible to fine-tune BERT with minimal task-specific architecture modifications, by adding a single output layer, for a wide range of downstream tasks such as language inference and question answering, resulting in superior performance as compared to LSTM-based LMs.

BERT is composed of multiple transformer layers that are stacked on top of each other. It uses a pre-training approach to learn bidirectional representations of tokens in a sequence, and the pre-training task is based on a Masked Language Model (MLM), where 15% of the Wordpiece tokens [115] in the input text are randomly masked and replaced with a special *[MASK]* token. During training, 80% of the masked tokens are replaced with *[MASK]*, 10% with a random token, and 10% are not replaced, and the goal is to predict the original tokens from the masked ones. However, masking tokens can cause a semantic misalignment between pre-training and fine-tuning. To address this issue, BERT fine-tuning replaces the *[MASK]* token with the original token, so the model can learn to predict the actual token in downstream tasks. BERT is pre-trained on a large corpus of textual data from the BookCorpus dataset [177] and the English Wikipedia. It can be fine-tuned for a variety of tasks, such as question answering [45] and language inference [103], with just an additional output layer.

The input to BERT is a text sequence with a *[SEP]* token separating the sequences, and a *[CLS]* token and a *[EOS]* token are appended to the beginning and end of the sequence respectively. Due to self-attention in the transformer layers, contextual embeddings are learned for each token during the pre-training process. These contextual token embeddings have demonstrated strong empirical performance on tasks such as QA, natural language inference, and text classification [22]. Devlin et al. [22] employed the *[CLS]* token representation or averaged token embeddings as the sentence representation of the sequence for downstream application tasks.

2.5.2 Sentence-BERT

As a result of the impressive Natural Language Understanding (NLU) capabilities of BERT, it has been successfully applied in determining the semantic relatedness of two sequences of tokens. This is achieved by concatenating the two sequences with a *[SEP]* token, and then prepending a *[CLS]* token to the concatenated sequence. The resulting sequence is then input into the BERT [22] model to obtain contextual representations for each token. The contextual representation of the *[CLS]* token or the average embedding of the sequence is then fed into a multi-layer perceptron layer for the final classification.

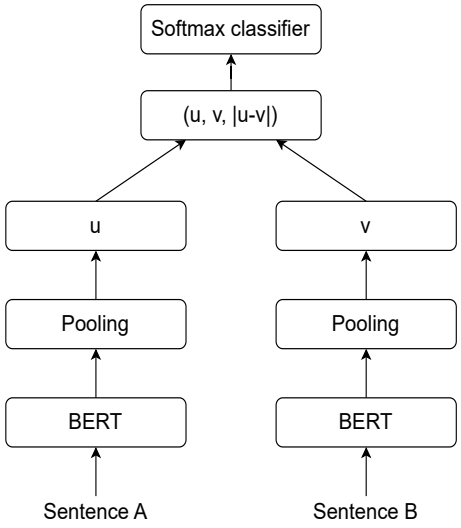

Figure 2.3: SBERT architecture proposed by Reimers and Gurevych [103].

Reimers and Gurevych [103] pointed out the limitations of the existing methodology for sentence similarity and proposed a new architecture called SBERT. The SBERT architecture employs a siamese network to learn semantically meaningful sentence embeddings. Siamese networks utilise the same network parameters (shared parameters) for different inputs. The sentences are first passed through a BERT layer to obtain contextual embeddings. These embeddings are then pooled using methods such as max-pooling, mean-pooling, or the *[CLS]* token embedding to obtain sentence representations. The SBERT architecture uses sentence embeddings and their difference to classify if two sentences are semantically related as illustrated in 2.3. The final classification layer can be replaced with a layer that maximises the cosine similarity between the two sentences if they are related. The framework was pre-trained on the Natural Language Inference (NLI) dataset [139, 15] to generate high-quality sentence embeddings that can be fine-tuned for downstream tasks.

In their work, Reimers and Gurevych [103] demonstrated that the SBERT architecture significantly improves the efficiency of semantic search compared to BERT. They reported that the search time for finding the most similar sentence pair in a collection of 10,000 sentences is reduced from 65 hours with BERT to the computation of 10,000 sentence embeddings, which takes approximately 5 seconds with SBERT. The cosine similarity computation between these embeddings takes around 0.01 seconds. These findings highlight the potential of the SBERT framework in enabling efficient and accurate semantic search.

2.5.3 Generative Pre-trained Transformer

BERT and SBERT architecture learn the contextual representation of tokens, and such representation can be utilised for NLP tasks such as text classification and QA. These LMs are encoder-only models as they do not have a decoder component and hence are not suited for generation-based tasks such as machine translation or dialogue generation. In order to overcome the limitations of existing LMs, particularly in the context of generating natural and coherent text, Radford and Narasimhan [99] proposed the Generative Pre-trained Transformers (GPT) framework. GPT is a stacked architecture composed of multiple layers of decoder-only transformer blocks and is

a decoder-only LM. The authors introduced a two-stage training process for adapting the pre-trained model to perform various NLP tasks. Firstly, the model is trained on a large corpus of unlabelled text for learning the initial neural network parameters. Then, it is fine-tuned discriminatively on labelled task-specific datasets to adapt the pre-trained model to specific tasks.

The GPT framework proposed by Radford and Narasimhan [99] consists of a two-stage training process. In the first stage, the GPT model is pre-trained on the BookCorpus dataset [177] using the standard language modelling objective. The pre-training dataset is composed of long contiguous pieces of text, allowing the model to learn to condition on long-range textual information. In the second stage, the pre-trained model is discriminatively fine-tuned on task-specific annotated datasets using specialised inputs. The authors demonstrated the strong empirical performance of GPT on various NLP tasks such as NLI, QA, commonsense reasoning, semantic similarity, and text classification.

Radford et al. [100] proposed the GPT-2 architecture, which is an extension of the GPT model, and has been widely adopted for various NLP applications due to its strong zero-shot learning capabilities. GPT-2 is similar to GPT in terms of architectural design, except for the layer normalization, which is moved to the input of each sub-block. Additionally, a layer normalization is added to the output of the final self-attention layer. GPT-2 was pre-trained on a large corpus of web-based text data using the language modelling objective and demonstrated impressive zero-shot performance on several NLP tasks, including commonsense reasoning, machine reading comprehension, summarization, and question answering. Its competitive performance on zero-shot learning has made it a popular choice for developing NLP systems.

2.5.4 Text-to-Text Transfer Transformer

Text-to-Text Transfer Transformer (T5)[101] is a neural network model based on the transformer architecture[130] designed for language modelling. T5 has transformer-based encoder-decoder components and has shown state-of-the-art performance on various NLP tasks, including text classification, QA, and machine translation.

T5, a neural network-based LM, utilises a two-stage training approach, where it employs a text-to-text learning scheme, receiving a sequence of text as input and generating a sequence of text as output, for training the model. The model is first pre-trained on the Colossal Clean Crawled Corpus (C_4) dataset [101], consisting of large amounts of clean English text, using a denoising objective, wherein 15% of the tokens in the input sequence are randomly dropped, and the model is trained to predict these dropped tokens. This setup is learned by minimising the standard language model loss as described in Equation 2.22. The denoising object enables T5 to learn robust representations of the language as it forces the model to learn robust representations by making it focus on relevant contextual information in order to predict the masked tokens.

The second stage of pre-training aims to adapt the pre-trained T5 model to perform specific NLP tasks, such as QA or machine translation. The T5 model is initialised with the pre-trained weights and is then trained on a task-specific dataset by minimising a task-specific objective, using a smaller learning rate compared to pre-training to avoid overfitting. It is worth noting that while the input to the T5 model may differ between tasks, the input and output remain in the form of text sequences. To further enhance performance, some task-specific modifications may be applied to the model's architecture or input/output format.

The T5 language model has demonstrated impressive performance across multiple NLP tasks. Leveraging the C4 dataset, T5 exhibits superior text generation capabilities with high quality and diversity. These results showcase the effectiveness of transfer learning and the potential benefits of large-scale pre-training for enhancing natural language processing.

2.5.5 Bidirectional and Auto-Regressive Transformer

Bidirectional and Auto-Regressive Transformer (BART) [62] is a sequence-to-sequence model that serves as a denoising autoencoder for a wide range of downstream tasks. Its encoder and decoder networks are based on transformer architecture. BART's encoder network is a bidirectional encoder (similar to BERT), while the decoder is a left-to-right decoder (similar to GPT). The model is pre-trained using negative log-likelihood optimization of the original document. BART is capable of handling a broad range of tasks due to its robust architecture.

BART is a neural network based LM that utilises a two-stage training process similar to T5 for adapting itself to specific tasks. BART is trained using a novel approach of document corruption followed by reconstruction. During the first stage, BART is pre-trained on a large corpus of text using a document-level noise injection method, where the model is trained to reconstruct the original document from its corrupted version. This is done by minimising the reconstruction loss, which is the cross-entropy between the decoder's output and the original document. Unlike other denoising autoencoders that are specific to certain types of noise, BART can handle any type of document corruption. If all information about the source document is lost, BART becomes equivalent to a language model. The use of document-level noise injection and reconstruction provides BART with the ability to generate high-quality text with a strong performance on various NLP tasks.

The second stage of task adaptation involves the use of the annotated dataset. Lewis et al. [62] highlighted the strengths of BART on text generation tasks such as machine translation, summarisation, dialogue and abstractiveQA, thus elucidating the benefits of BART in addressing NLP tasks.

2.5.6 Neural Conversational Architectures

Neural network architectures have found effective applications in modelling conversational interactions. For instance, Zhang et al. [163] introduced DialoGPT, a neural model designed for conversation generation. DialoGPT adopts the architecture of GPT-2 [100] as the basis for generating responses. It undergoes fine-tuning using a vast dataset of Reddit conversations, encompassing over 147 million dialogues and 1.8 billion words. This approach demonstrated strong empirical performance in generating responses. Furthermore, Adiwardana et al. [3] introduced Meena, a neural architecture with 2.6 billion parameters tailored for open-domain conversation generation. Similar to DialoGPT, Meena relies on transformer blocks as fundamental building components. While DialoGPT is a decoder-only model, Meena is an encoder-decoder model equipped with 13 transformer blocks in both the encoder and decoder networks. It is fine-tuned on a massive dataset of open-domain social media conversations, spanning more than 40 billion words. Additionally, Roller et al. [105] introduced BlenderBot, another neural response generation model based on the transformer architecture. BlenderBot is an encoder-decoder network

fine-tuned using the pushshift Reddit corpus [11] for open-domain dialogue generation. These advancements in neural conversation architectures have inspired researchers to leverage neural models in the development of chatbots and dialogue systems.

2.6 SUMMARY

This chapter provides the fundamental concepts essential for understanding the ideas proposed in this thesis. It begins by introducing the different types of conversations typically handled by chatbots. Then, the basics of KGs are introduced, which are necessary to grasp the concepts presented in this thesis about structured knowledge-grounded chatbots. After that, the workings of neural networks, which form the basis of various methodologies proposed in this thesis, are explored. Finally, different Language Models (LMs) utilised to capture the semantics of conversations in this thesis are introduced.

3 RELATED WORK

This chapter presents an overview of the research conducted in the field of knowledge-grounded chatbots. It begins by introducing seminal works in knowledge selection from structured sources such as KGs in Section 3.1, primarily in the context of conversational recommendation systems. Section 3.2 provides a brief overview of various methodologies proposed for selecting relevant knowledge from unstructured sources in the QA components of a chatbot. Subsequently, Section 3.3 discusses different techniques suggested for achieving conversational explainability in a chatbot. Finally, Section 3.4 delves into approaches for generating informative responses using both structured and unstructured sources of knowledge in a chatbot.

3.1 KNOWLEDGE SELECTION FROM STRUCTURED SOURCES

One of the key challenges in building chatbots is improving their capabilities to have effective and engaging conversations with users. This entails supplementing the model with knowledge sources, thus enabling the system to introduce relevant facts and entities in the conversation. In this section, we review the utilisation of structured knowledge in the form of KG in CRSs, exploring the different neural architectures employed for an appropriate recommendation.

Li et al. [64] proposed the ReDial dataset, a large-scale conversational recommendation dataset helpful in training deep learning models. Each dialogue is a conversation between a recommender and a recommendee, wherein the recommendee is seeking a movie recommendation. The dataset contains more than 10,000 dialogues focused on movie recommendations. Additionally, they proposed the first deep learning architecture suited for the task of conversation recommendation. They used the Hierarchical Recurrent Encoder-Decoder architecture [112] to encode the dialogue history for capturing contextual information across multiple turns. The integration of contextual information over multiple turns contributed to a richer understanding of user preferences. This contextual information is then used to generate the recommendation seeker's sentiment about the movies. As sentiments contain vital information about user preferences, an autoencoder is utilised to leverage the sentiments for predicting the users' ratings of all the movies present in the database. The proposed framework utilised these predicted ratings as a discerning criterion for recommending appropriate movies in the response.

Zhou et al. [175] introduced the TG-ReDial dataset, wherein the conversation recommendation dialogues incorporate topics threads, leading to a natural transition of the dialogues into a movie recommendation dialogue. In contrast to the ReDial dataset, the TG-ReDial dataset incorporates topic threads, which consist of sequences of topics observed within the dialogue context. These topic threads serve to direct the content progression within each conversation. In addition, they proposed a framework for recommending movies to users. The proposed recommendation module uses BERT representations of the dialogue history and historical item interactions to

construct user representations. As outlined in Section 2.5.1, BERT is an efficient language model for capturing contextual information. In this scenario, the BERT representations are an effective choice for capturing the contextual information in the dialogue history. The recommendation module computes the similarity between the user representation and the item representations to recommend relevant items to the user. A unique aspect of the TG-ReDial dataset is that it is presented in Chinese, thereby facilitating research in CRS in Chinese alongside English.

Chen et al. [18] grounded the ReDial dataset on the DBpedia [8] KG for supplementing the recommendation module with external knowledge. DBpedia KG has factual relationships between entities that can enable a richer understanding of the dialogue context. Additionally, they suggested the use of information about both item and non-item entities in the KG. Item entities denote the entities being recommended, which, in the case of the ReDial dataset, correspond to the collection of movies. On the other hand, non-item entities originate from the DBpedia Knowledge Graph and encompass entities other than movies, such as writers, actors, or directors. The authors utilised Graph Neural Networks (GNNs) [109] which leverages the KG structure for computing entity representations. The representations of the item and non-item entities are then utilised for computing the user representations. Self-attention is employed on both item and non-item entity representations to derive user representations, encapsulating the user's preferences within the conversational context. Finally, the dot-product similarity of the user representation is computed with the item representation for recommending appropriate movie entities.

Zhou et al. [174] observed the lack of the utilisation of commonsense information in prior works and proposed KGSF, a framework leveraging the benefits of both fact-based KG (e.g. DBpedia) and commonsense KG (e.g. Conceptnet [118] for the recommendation and response generation task. Tokens existing in the dialogue and aligning with entries in the ConceptNet KG are referred to as "concepts." Unlike the DBpedia KG, which predominantly encodes factual associations among entities, the ConceptNet KG encompasses more comprehensive common-sense relationships and affiliations spanning emotions, beliefs, and everyday knowledge. KGSF focuses on increasing the mutual information between entities and concept words occurring in the same context. Mutual information maximisation focuses on enhancing the information shared between the DBpedia entities and the Conceptnet entities present in the dialogue history, thus enabling the learning of semantically rich embeddings. These enriched embeddings are used for user representation using the self-attention mechanism, which is then used for recommendation.

The utilisation of structured knowledge for CRS encouraged researchers to explore the utilisation of unstructured knowledge as an external source of knowledge for the task. Lu et al. [77] proposed RevCore, a deep learning architecture employing item reviews during recommendation in addition to leveraging a structured KGs. As reviews capture the inherent bias of the product and can help capture user preferences, the researchers showed improved performance over different datasets. To address the problem of cold-start in CRSs, [67] proposed to learn user preferences from semantically similar conversations using collaborative filtering. The approach focuses on enhancing user and item representations by incorporating item-aware and user-aware information respectively. An interactive user-item graph is constructed to incorporate popularity-aware information for item representation learning, and a retrieval-enhanced method is proposed to model user preferences.

3.2 KNOWLEDGE SELECTION FROM UNSTRUCTURED SOURCES

Traditional QA systems used lexical parameters such as question classes (e.g., what, who, etc.) detection or bag-of-words (BoW) matching between the question and context to fetch the appropriate answers [24]. QA systems based on IR techniques and knowledge graphs were studied with the introduction of IBM Watson [29] *DeepQA*. The rich development of deep learning architectures led to the advent of QA systems that were based on neural models such as RNN, and Long-Short Term Memory (LSTM) [71] – depicting decent performance.

The creation of QA oriented datasets like SQuAD [102], NewsQA [126] and WikiQA [152] spurred research interest in this domain. SQuAD [102] comprises a series of more than 100,000 questions that have been generated by crowdworkers based on a set of Wikipedia articles. For each question, the answer can be found within a specific segment of text from the corresponding reading passage. Similarly, NewsQA [126] dataset is a set of more than 100,000 human-generated question-answer pairs. The crowdworkers generate the questions from a set of over 10,000 news articles with the answers being presented as spans of text within the articles. WikiQA [152] is a set of user questions sampled from search engine queries. The questions are linked to a Wikipedia document for answer retrieval. It is interesting to note that questions in the WikiQA dataset may or may not contain an answer. HotpotQA [153] introduces a new challenge in the task of Machine Reading Comprehension (MRC). Reasoning over multiple documents is required for answering the questions in the dataset. Kwiatkowski et al. [55] The dataset consists of real queries made to the Google search engine, which are anonymised and aggregated. Each query is presented to an annotator along with a Wikipedia page from the top 5 search results. The annotator then provides a long answer, typically a paragraph, and a short answer, which can be one or more entities if present on the page, or marks null if no answer is present.

Fueled by the growth of language models and their enhanced performance in several NLU tasks, most QA systems predominantly rely on language models fine-tuned on QA data [164], e.g., RoBERTa-SQuAD, ELECTRA-SQuAD,[1], etc. Models like SpanBERT [45] were specifically pre-trained (using a specialised training loss function) to be geared towards QA. SpanBERT involves masking contiguous spans of tokens, rather than individual tokens, and training the model to predict the entire masked span without relying on the individual token representations within it. This results in more effective modelling of contextual relationships between tokens in a sentence. To mitigate the challenge of domain adaptability, domain-specific language model like BioELECTRA++ was explored for bio-medical QA [90], wherein the model was utilised for tasks such as answer sentence reranking, summarisation and QA, while adversarial training and multi-task learning was also studied [63]. In fact, a multi-stage QA pipeline combining retrieval, neural ranking and transformer-based models were proposed for closed-domain conversational agents [110] for technical QA [158] and legal QA [44]. Attention mechanism to domain-specific contexts were also incorporated for expanding QA systems to restrictive domains [111]. Synthetic question generation for fine-tuning a neural architecture-based QA system was recently shown to produce good results [60]. The popularity of QA systems have enabled its growth in multi-lingual settings [75] as well as for multi-modal datasets [122]. Observe, in general, almost all domain-specific QA techniques rely on annotated training data

1 huggingface.co/models?pipeline_tag=question-answering

3.2.1 Synthetic QA generation

The use of synthetic dataset generation for training has been posited in EQA. Alberti et al. [5] follow a two-step process for synthetic QA dataset generation, where they train a BERT [22] model to extract answers from the context passage. Thereafter they train another BERT model that generates the question using the answer extracted from the previous step and the passage context. The encoder stack of BERT is treated as a left-to-right language model and is then utilised for the generation task. Chan and Fan [17] improved upon the question generation by leveraging sequential neural architectures built on top of BERT. While BERT models have been largely used to generate synthetic data, Puri et al. [98] used a GPT-2 [100] model to generate QA pairs and used BERT model to filter QA pairs using round-trip consistency. Additionally, Lu et al. [76] utilised the generation of a synthetic dataset for neural passage retrieval. Deng et al. [21] augmented a golden labelled dataset with an additional synthetic dataset for the ranking of question-and-answer sentence pairs. As outlined in Section 7.2.1, *SEDAN* uses a T5 model to generate QA pairs and uses this synthetic dataset to adapt an AS2 model to a closed-domain. Shakeri et al. [114] explored the generation of QA pairs in a multilingual setting. They leverage the multilingual T5 model trained only on English QA datasets for generating synthetic QA pairs in Arabic, Russian, Hindi, Chinese and Spanish. Similarly, Riabi et al. [104] first generated synthetic QA pairs in English and then used a translation tool for translating the synthetically generated QA pairs to the target language for zero-shot multilingual QA. Pappas et al. [93] proposed domain adaptation of QA models in the bio-medical domain by augmenting the training dataset with synthetically generated QA pairs. Kramchaninova and Defauw [52] explored the domain adaptation of QA models in a multilingual setting. They augmented the non-synthetic QA data with a synthetic dataset and fine-tuned a multilingual BERT model for QA.

3.2.2 Answer Sentence Selection

Neural network models have been studied extensively in search, information extraction, text reranking [36], AS2 task [58, 37, 57] and for measuring textual semantic similarity [22, 72, 103]. In the context of AS2, researchers have employed Convolutional Neural Networks (CNNs) and RNNs to learn the semantic relationship between text pairs and in turn rank textual context relevant to user needs. Severyn and Moschitti [113] employed a CNN network to learn a similarity function between pairs of texts, while Tan et al. [123] proposed modelling context information with hierarchical gated recurrent neural networks.

While such models perform well, transformer [130] based AS2 models [37, 58, 57] have been shown to significantly outperform CNN and RNN based models. Garg et al. [37] proposed fine-tuning of transformer-based language models for the AS2 task. They released a large-scale AS2 dataset and showcased the need for a two-step fine-tuning process for domain adaptation in AS2. Specifically, they suggest the first step of fine-tuning on a general-purpose AS2 dataset to adapt transformer-based language models for the AS2 task. The authors recommend a second fine-tuning stage on a domain-specific dataset for effective domain adaptation. Yoon et al. [155] proposed the Propoagate-Selector framework for detecting sentences that can potentially answer a question. They design a graph structure with each node representing a sentence. A GNN is trained over the constructed graph for detecting the supporting sentences. Lauriola and Moschitti [58] suggested the use of a document-level global context in addition to the local context

used by Garg et al. [37] to improve the performance of transformer-based AS2 model. They showcased that local context helps in resolving implicit references, while global context enables the understanding of the content or document information. Liello et al. [66] introduce novel pre-training objectives for sentence-level transformers that take into account paragraph-level semantics within and across documents. The proposed objectives are designed to enhance the performance of transformers for AS2 and reduce the need for large labelled datasets. Gabburo et al. [35] propose a two-stage approach that leverages an AS2 model and a synthetic answer generation model for AS2. Given a question, the AS2 model is used to rank answer candidates without explicit supervision, and the top-ranked answer is used as the target for the answer generation model. The answer generation model takes the question and the top-k ranked answers from the AS2 model as input. This is then used to re-rank the answer sentences by tuning the AS2 model.

3.3 CONVERSATION EXPLANATION USING KNOWLEDGE GRAPHS

KGs are a rich source of structured information and contain semantic relationships between entities. Paths over a KG convey semantic information changes when moving from one entity to another. Paths over KG have been utilised for providing explanations to recommendations [33, 134], dialogues [84, 46] and QA [154, 146, 145] and for fact checking [131].

In recent years, the use of reinforcement learning [47] to address reasoning tasks using KGs has received significant attention. Reinforcement learning is a learning methodology wherein an agent learns a task by interaction with the environment. For reasoning tasks, the environment is generally defined by the properties of the KG. Specifically, Lin et al. [68] proposed a reinforcement learning-based approach to enable an agent to identify links within KG. Similarly, Wang et al. [134] used reinforcement learning to generate walk paths over a KG to provide explanations for product recommendations. They trained a reinforcement learning agent to traverse a KG and reward the agent when it arrives at an appropriate product in the KG. Zhao et al. [165] improved upon the previous work by training a reinforcement learning agent for the reasoning task. They utilised true demonstration and rewarded the agent if the path was similar to the ground truth path for explainable recommendations. Furthermore, methodologies [33] have been developed to ensure fairness in explainable recommendations over KGs.

Young et al. [156] attempted to integrate a large scale KG into an end-to-end dialogue system. Chen et al. [18] augmented user chats with a KG for recommending movies to users in a conversation recommendation setting. They utilised GNN for learning user preferences and thereafter recommend movie entities to users. Following this, Zhou et al. [174] proposed an approach that combines a commonsense KG (e.g. ConceptNet [118]) with a factual KG (e.g. DBpedia [8]) to learn user preferences based on their movie-related sentiments. They utilised this setup for recommending movie entities from the factual KG to users. Wang et al. [135] adopt the prompt tuning [16] paradigm to tackle the conversation recommendation problem. They propose a knowledge-enhanced prompt approach that leverages a pre-trained language model to generate suitable prompts for recommending products to users from a factual KG. Though successful in retrieving suitable entities or facts from the KG, these systems fail to provide explainability to the recommendations in the response text.

Such limitations encouraged explainable conversation reasoning using external knowledge. Liu et al. [73] develop the problem as a Partially Observable Markov Decision Process (POMDP) and use reinforcement learning for training the agent to generate KG paths. Moon et al. [84] posited a KG path-parallel-dialogue corpus along with DialKG Walker (DKGW) model, a recurrent decoder model to generate the KG path for a response entity selection. DKGW utilises an LSTM encoder for encoding the dialogue context. The model encodes the KG entities using TransE [14] embeddings. The dialogue representation along with the representation of entities in the dialogue history is then sent to a RNN decoder for generating entities at each time step. The sequence of entities generated by the decoder forms a path over the KG enabling conversation explainability. Jung et al. [46] proposed AttnIO, which leverages GNN using attention flow to generate KG entity paths. AttnIO encodes the dialogue history using a transformer-based pre-trained language model, and this contextual information is infused in the message passing network of the GNN for node scoring. The path with the highest score is selected to be the path for explainability. While novel, DKGW does not explicitly utilise the graph structure during model training. On the other hand, the performance of AttnIO [46] relies on the node sampler during training. AttnIO becomes computationally expensive due to dialogue-specific GNNs (both during training and inference) as the model concatenates the dialogue embedding to the node embeddings while propagating attention scores. Ni et al. [88] proposes a hierarchical model for learning goal planning by taking into consideration local and global user preferences in a dialogue. They model user preferences using paths over a KG and introduce HiTKG, a hierarchical transformer-based graph walker that utilises multiscale inputs to accurately and flexibly predict KG paths for conversation explainability. Similarly, Tuan et al. [128] utilises a pre-trained transformer model to incorporate reasoning capabilities into dialogue models.

3.4 RESPONSE GENERATION FROM STRUCTURED AND UNSTRUCTURED KNOWLEDGE SOURCES

Knowledge-grounded chatbots focus on generating informative responses pertinent to the dialogue context. One of the early studies to demonstrate the efficacy of RNNs in response generation was presented by Sordoni et al. [117], wherein they trained a RNN network on Twitter data for dialogue response. Serban et al. [112] introduced a hierarchical recurrent encoder-decoder framework for generating responses and demonstrated its strong performance compared to other competing methodologies. Dusek and Jurcícek [26] introduced a sequence-to-sequence model based on LSTM to adapt to the user's conversational style, incorporate contextual information, and generate context-dependent responses in a spoken dialogue system setting. With the success of the transformer architecture, researchers explored transformer-based models for dialogue understanding and response generation [166]. Following this, Roller et al. [105] suggested multiple blending skills that would enable a chatbot to generate engaging, knowledgeable and personalised responses to users. Additionally, they suggested careful considerations of the choice of hyperparameters and generation strategies for fluent response generation. Zhang et al. [163] proposed Dialo-GPT for open-domain response generation. The authors adapted a GPT-2 [100] based model trained as an autoregressive model on large-scale dialogue corpus for human-like

dialogue response generation. Similarly, Kulhánek et al. [54] proposed AuGPT, a GPT-2 based model for task-oriented dialogue by leveraging synthetic back-translated data.

The development of chat datasets grounded in knowledge of high quality has generated interest in the field. Dinan et al. [23] proposed the Wizard of Wikipedia dataset wherein dialogues are grounded on Wikipedia articles. Each conversation in the Wizard of Wikipedia dataset has two participants engaging in a chit-chat conversation about different topics of interest. One of the participants plays the role of a *apprentice* and the other participant plays the role of a *knowledgeable expert*. The role of the *apprentice* is to be a curious learner and ask for detailed information about the topic of interest. On the other hand, the role of the *expert* is to utilise information from Wikipedia and supply knowledgeable information in the conversation. Similarly, Zhou et al. [173] presented the CMU-DoG dataset, a dataset for document-grounded conversations, where the conversations are centred around the content of Wikipedia documents about popular movies. Moghe et al. [82] present Holl-E, a novel dataset of movie conversations, where each response is generated by incorporating or modifying sentences from unstructured sources of background knowledge, such as movie plots, comments, and reviews.

In contrast, similar to the OpenDialKG dataset [84], the DuConv dataset [141], which contains conversations between users that are guided by a KG. This dataset serves as an additional resource for facilitating further exploration and development in the area of KG grounded dialogue. The DuConv dataset's dialogues are rooted in a blend of both structured and unstructured knowledge sources. Following a similar setup to the construction of the Wizard of Wikipedia dataset, each conversation within the DuConv dataset involves two participants: a designated *leader* and a *follower*. The *leader* initiates the conversation on a specific subject, manages topic transitions throughout the dialogue, and employs information from knowledge sources for generating responses. Conversely, the *follower* assumes the role of responding to the ongoing dialogue context without drawing upon external knowledge. Zhou et al. [172] proposed the KdConv dataset, wherein conversations are grounded on both structured and unstructured knowledge while being spread over multiple domains. Crowdworkers are to respond to the conversation context using information either from a KG or from an article about an entity. While both DuConv and KdConv datasets are informed by a fusion of structured and unstructured knowledge sources, DuConv dialogues are constrained to encompass merely two distinct topics. In contrast, KdConv dialogues exhibit a broader spectrum, spanning from one to four topics. Furthermore, while the DuConv dataset pertains solely to the *film* domain, the KdConv dataset spans across the *film*, *music*, and *travel* domains. Consequently, the response generation task within the KdConv dataset assumes an increased level of complexity and challenge. This introduced an additional challenge of fetching appropriate information from multiple sources of knowledge.

Dinan et al. [23] noticed that models trained on open-domain data suffer from hallucinations and produce factually inconsistent responses. This necessitated the need for knowledge-grounded dialogue systems. They proposed a novel transformer memory network for knowledgeable response retrieval. The transformer memory network encodes the knowledge elements and the dialogue context using the transformer network. An attention layer is applied to the knowledge representations and the context representation to construct the dialogue representation. The candidate responses are encoded using a separate transformer network and finally, the response with the highest dot-product with the dialogue representation is chosen to be the candidate response. Following this, Zhao et al. [167] proposed the joint unsupervised learning

of the knowledge selector and the pre-trained language model using a reinforcement learning approach. They showcased that joint training of the knowledge selector and the response generator is beneficial in producing coherent and informative responses. Fu et al. [32] proposed a personalised approach to knowledge-grounded dialogue by incorporating the personal memory of the interlocutors, which can influence the selection of knowledge in a dialogue. The study investigated the impact of personalisation on the performance of knowledge-grounded chatbots by variationally modelling the preferences of users.

While prior works focused on grounding dialogue systems on unstructured sources of knowledge, Zhu et al. [176] suggested grounding dialogue systems on structured sources such as KGs. They suggested the GenDS framework, which uses a knowledge retriever for fetching appropriate knowledge and a generator for producing a response conditioned on the dialogue history and knowledge facts. The response generator is entity-representation agnostic allowing GenDS to produce both seen and unseen entities in the response text. Zhou et al. [171] incorporated a vast commonsense knowledge graph (KG) into an open-domain chatbot to enhance language comprehension and generate responses. They employed graph representations instead of entity representations, treating the knowledge triples as a graph to better interpret the semantics of an entity based on its neighbouring entities and relations. Zhou et al. [170] posited the EARL framework for knowledgeable response generation. EARL is designed to be entity-agnostic and uses both the dialogue context and relationships within a knowledge graph (KG) to understand semantics and information flow. To achieve this, EARL masks entity tokens and learns to capture the semantic relationships between the context and KG relations. As no entity-specific embeddings are learned, the framework can effectively introduce new and unseen facts in the responses.

Another line of work has been exploring prompt-tuning methodologies for response generation. In these prompt-based models, tokens are sent as a context to language models for response generation without fine-tuning the parameters of the language model [61]. Consequently, a portion of parameters undergo tuning, while the intrinsic parameters of the underlying LM remain unaltered. This approach streamlines model learning, demanding less computational resources, and enables efficient domain adaptation, as task-specific learning involves a relatively small number of parameters, typically in the order of a few thousand. Zheng and Huang [168] suggest using few-shot examples as prompt tokens for knowledge-grounded response generation. The work by Liu et al. [74] introduced a multi-stage dialogue prompting framework that comprises two stages: the first-stage prompting for knowledge generation and the second-stage prompting for response generation. In these prompt-based models, the dialogue context and learnable prompt tokens are sent to language models for knowledgeable dialogue.

A major drawback of such models is the inability to utilise structured and unstructured knowledge in unison. Zhou et al. [172] proposed using a memory-based Hierarchical Recurrent Neural Network encoder-decoder network for grounding dialogue agents on both knowledge sources. The encoder-decoder network utilises the knowledge elements from both structured and unstructured sources along with the conversation context as the output of the encoder network. This encoder representation is then sent to the decoder network for final response generation. Following this, Wang et al. [133] suggested RT-KGD, a novel framework utilising dialogue transition for knowledgeable dialogue generation. They encode elements from the KG using TransR [69] embedding. The KG embeddings are then sent to a BART-based encoder along with the unstructured knowledge tokens and the conversation context. The output of

this encoder network is then sent to a BART-based decoder network for response generation. RT-KGD does not represent knowledge elements and the dialogue history in the same semantic vector space leading to an information gap between the dialogue context and the knowledge elements.

3.5 SUMMARY

In this chapter, we present an overview of different works pertaining to knowledge-grounded chatbots. The chapter encompasses literature on knowledge retrieval from structured sources of knowledge (e.g. KGs), knowledge retrieval from unstructured sources of knowledge (e.g. Wikipedia or domain-specific documents), achieving explainability, and generating informative responses in chatbots. First, we discuss seminal works on retrieving knowledge from structured knowledge sources in conversational recommendation systems. Thereafter, we highlight different methodologies for knowledge retrieval from unstructured sources of knowledge in the context of QA components of a chatbot. We delve into techniques suggested for achieving the explainability of conversations undertaken by a chatbot. Finally, we discuss different methodologies for generating informative and fluent responses leveraging both structured and unstructured knowledge sources in a chatbot.

4 KNOWLEDGE SELECTION FROM STRUCTURED SOURCES

Chatbots have become ubiquitous parts of our lives in the form of virtual assistants (e.g. Siri, Alexa, Google Assistant). Chit-chat conversations, question-answering and conversational recommendations are a few use-cases where chatbots have been shown to be quite effective. Conversational Recommender System (CRS) focus on the task of suggesting products to users based on the conversation flow. The use of external knowledge in the form of KGs has been shown to improve the performance of chatbots in a conversational recommendation setting [18, 174]. Structured information from KGs aids in enriching conversational recommendation systems by providing additional information such as closely related products, textual descriptions of the items and user-item interaction [18, 174]. However, KGs are inherently incomplete as they do not contain all the factual information present on the web. This leads to the systems learning incomplete information about different items or entities in the KGs. Additionally, KGs contain millions of facts within themselves. When working on a specific conversation, KGs in their entirety contribute towards extraneous information and noise since not all facts are relevant to the conversation. This phenomenon raises questions about the impact of KG size and vocabulary on the system's performance in general and on the performance of the *structured knowledge retriever* (Figure 1.2) in particular. In this chapter, we study several subgraph extraction methods and compare their performance impacts across the recommendation task in conversational recommendation systems. This enables the understanding of the impact of KG size on the knowledge selection performance of chatbots. We incorporate pre-trained embeddings from the subgraphs along with positional embeddings in our models. Extensive experiments showcase the effectiveness of our proposed approach by improving upon the state-of-the-art methodology on multiple metrics for the conversation recommendation task.

4.1 INTRODUCTION

CRSs are goal-oriented chatbots focusing on recommending products to users through multi-turn conversations. These conversations facilitate the interactions between the user and the recommender system. CRSs hold enormous potential in the e-commerce industry wherein users can be recommended products based directly on the understanding of their requirements. However, users are not always entirely aware of their preferences while purchasing products. A CRS enables the user to make an informed decision while purchasing a product by taking into consideration the information about different products and matching them to their needs [64]. This is extremely useful in situations of high-involvement products. By understanding the context and learning the user's preference, CRS can suggest products to the users which in turn will lead to

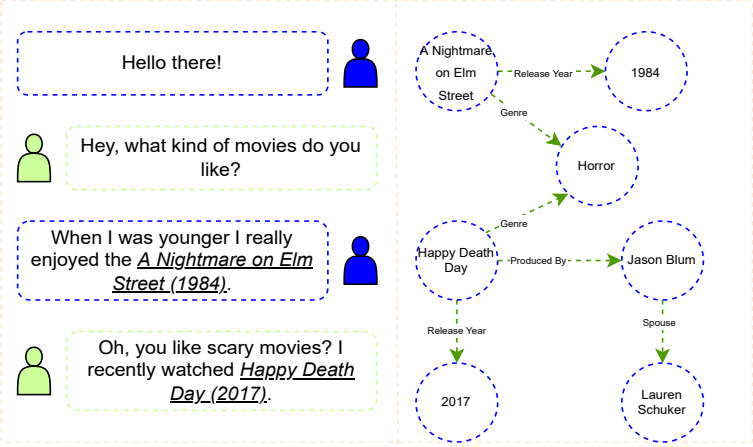

Figure 4.1: An example conversation wherein a user is seeking movie recommendations from the chatbot. Utilising the semantics of the conversation, the chatbot recommends the movie *Happy Death Day* to the user.

higher consumer satisfaction and low buyer's remorse.

The two central components of CRS are the *recommender system*, the *dialogue system* [18]. The *recommender system* is aimed at retrieving a subset of items that meet the user's interest from a larger set of items. The *dialogue system* focuses on generating appropriate responses conditioned on the conversation history and user's item preferences. To draw a parallel, the *recommendation system* and the *dialogue system* resemble the *knowledge retriever* and the *response generator* respectively as illustrated in Figure 1.2.

More recently, work on leveraging information from KGs have been gaining ground. KGs contain factual information about real-world entities in a structured format. Such external knowledge helps in adapting to a specific domain quickly as they contain rich information about products and their features. This knowledge helps the recommendation system utilise additional information about products, leading to better suggestions. Leveraging such graphs helps in making a better recommendation as supplementary information is made available to the system in addition to the conversation context and user behaviour. Considering the example showcased in Figure 4.1, where the user is interacting with the chatbot to request a movie recommendation. After inferring users' preferences about *scary movies* and the previously watched movie "*A Nightmare on Elm Street (1984)*", the chatbot utilises the KG information to recommend "*Happy Death Day (2017)*". During the recommendation process, the chatbot first understands the user preference from the conversation history and then utilises the external KG for recommending the appropriate movie. One should notice the presence of extraneous edges in the KG. Given this conversation, the KG edge *(Jason Blum, Spouse, Lauren Schuker)* is irrelevant considering the conversation being processed. This further strengthens the need for appropriate subgraph extraction methodologies, thereby leading to improved knowledge selection from the KGs.

We improve recommendations provided to users by incorporating KGs in chatbots for movie recommendation. However, to leverage the benefits of KGs, there is a need for constructing appropriate subgraphs for the domain, thereby reducing the amount of redundant information made available to the system. Subgraphs that are rich in domain information containing lower

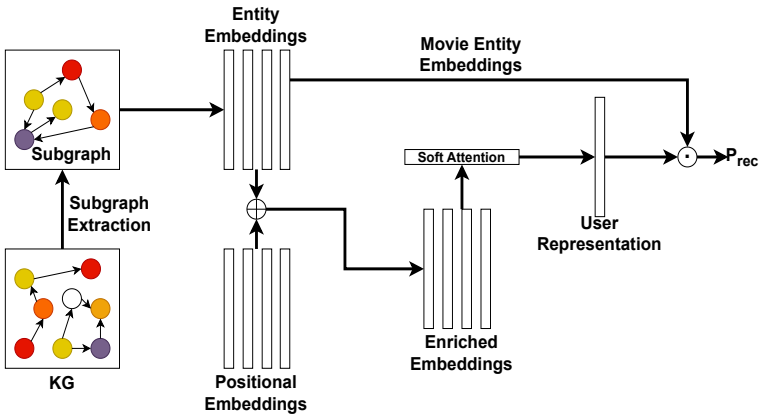

Figure 4.2: Overview of our model. A subgraph is constructed from the KG. Entity embeddings are learned on the extracted subgraph. Entity embeddings belonging to a particular user are enriched with positional embeddings ($\oplus$ represents elements-wise addition) and then passed through a soft-attention layer to represent the user. The score of each entity is calculated by taking a dot-product ($\odot$ represents the dot-product of two vectors) of its embedding with the user representation. Finally, the probability of each entity is calculated by passing the scores of all entities through a softmax layer.

amounts of noise are desirable. We, therefore, study the benefits of subgraphs created using N-hop and PageRank [91] approaches to analyse which subgraphs are best suited for the task at hand.

In this work, we build upon the work of Knowledge-Based Recommendation Dialog (KBRD) [18]. The authors extract movie entities from dialogues and utilise information from a KG to suggest movies to users. We incorporate pre-trained entity embeddings and make use of positional embeddings to improve the performance of the system. Our contributions towards this study are as follows:

- We conduct extensive experiments on different subgraphs extracted from DBpedia [8] to show that information contained in them is different and that there is a need for an optimal subgraph creation technique.

- We show that using pre-trained entity embeddings supplemented with positional embeddings allows the model to learn better entity representation for the recommendation.

4.2 METHODOLOGY

To address the recommendation task, we begin with constructing a subgraph from the KG. We then learn entity embeddings on the subgraphs thus constructed using the method described by Balazevic et al. [10]. The learned embeddings are used for user representation. Finally, we compute the similarity of the user with the movie entities. An overview of our model is shown in Figure 4.2.

Table 4.1: Properties of subgraphs extracted for the movie domain from DBpedia. As the number of hops increases, the number of nodes and edges increases while the density decreases. Subgraphs extracted using PageRank show a slight difference in their properties as a result of different personalisation values.

Subgraph	#Nodes	#Relations	#Edges	Density
2-hops	64,368	214	153,852	7.42e-05
3-hops	101,158	410	265,910	5.19e-05
5-hops	217,334	497	642,633	2.72e-05
PageRank	80,181	399	187,491	5.83e-05
PPR($\alpha = 0.9$)	68,005	321	162,326	7.02e-05
PPR($\alpha = 0.7$)	69,012	333	165,124	6.93e-05

4.2.1 Formal Problem Definition

A user u from the user set $\mathbb{U}$ is interacting with a CRS through a conversation $\mathbb{C}$ consisting of utterances $\{c_1, c_2, ..., c_{n-1}\}$. The CRS needs to select an item set I_t from the set of all available items $\mathbb{I}$ using the information available in $\mathbb{C}$.

4.2.2 Subgraph Creation

In order to perform analysis on the contributions of various subgraphs, we employ the use of N-hop neighbourhood and PageRank algorithms for subgraph extraction. In all of the cases, the initial set of seed nodes is the set of movie entities present in the ReDial dataset[64]. When extracting the subgraph using the N-hop method, we include all the edges and nodes reachable from the seed set by paths of length N. We consider cases when N=2, 3 and 5 starting from the initial set of seed nodes.

In contrast, the extraction of subgraphs using PageRank is a two-step process. An initial subgraph is extracted by considering 3-hop neighbours of the seed set. PageRank scores of the subgraph nodes are computed and the top-*k* nodes with the highest PageRank scores are considered for the second step. To extract the final subgraph, we consider the top-*k* nodes with respect to the PageRank values along with the movie nodes. For the purpose of this work, we select the top 30,000 nodes having the highest PageRank scores. We take the set of nodes thus extracted and their 1-hop neighbours to present the final subgraph. We also consider Personalised PageRank (PPR) [9] to extract subgraphs wherein each node is assigned a personalisation score. In our experiments, we distribute α between the set of movie nodes and $1 - \alpha$ among the rest of the nodes in the subgraph extracted after the first step. We consider the cases when $\alpha = 0.7$ and 0.9. Table 4.1 shows the graph properties of different subgraphs extracted. We analyse the number of nodes, edges, relations and the density of the graphs thus extracted. The density of a graph measures how interconnected the graph is and is defined as the ratio of the number of edges in a graph to the maximum possible number of edges in a graph. 5-hop subgraph contains the highest number of nodes and edges while having the lowest density. We learn entity embeddings for these subgraphs extracted for use in the model.

Table 4.2: Distribution (in %) of the distance of target nodes from the source nodes in different subgraphs. All the subgraphs show similar distributions with the probability of a target node being unreachable close to 0.18.

Subgraph	0	1	2	3	4	5	$\geqslant 6$	Unreachable
2-hops	9.82	4.64	21.11	14.68	29.94	1.15	0.47	18.16
3-hops	9.77	4.62	21.04	15.27	29.45	1.39	0.43	18.01
5-hops	9.70	4.58	20.92	15.52	29.38	1.67	0.28	17.92
PageRank	9.83	4.66	21.18	14.35	28.44	2.30	0.77	18.44
PPR($\alpha = 0.9$)	9.82	4.64	21.12	14.71	29.63	1.47	0.37	18.20
PPR($\alpha = 0.7$)	9.82	4.64	21.12	14.71	29.86	1.23	0.39	18.19

4.2.3 Entity Embeddings

To incorporate information from the relational graph into our model, we need to represent the entities as a d-dimensional vector. More specifically, each entity is represented as a $\mathbf{R^d}$ vector. In this work, the embedding of an entity $\mathbf{i}$ is represented by $\mathbf{e_i} \in \mathbf{R^d}$. Balazevic et al. [10] learn entity embeddings by solving link-prediction tasks in relational graphs. Since those embeddings are well suited for the task of link prediction, they contain relevant information for predicting new relations in an incomplete graph. Connectionist models such as those demonstrated by Schlichtkrull et al. [109], leverage the local neighbourhood of nodes to learn nodal embeddings. Although relational graph neural networks help in learning better entity representation for relational graphs, KGs do not contain all factual relations. As shown in Table 4.2, for all of the subgraphs, at least 18% of the target nodes are unreachable from the previously mentioned entities. Therefore, we choose the above-mentioned method for learning entity embeddings.

4.2.3.1 TuckER Embeddings

GNNs [143] allow the information to flow through a graph using connectionist models. However, such models do not allow information flow between nodes that are not connected through paths of any length. Since, in our case, many target entities are not connected to source nodes, we use embeddings learned through Tucker decomposition [129] for better entity prediction. The nature of the learned embeddings is such that it helps in link prediction between two entities.

Tucker decomposition decomposes a tensor into a smaller core tensor and several smaller matrices. Balazevic et al. [10] show that Tucker decomposition can be used to learn entity embeddings from relational graphs. They also showed that embeddings learned using this technique outperforms techniques such as TransE [14] or DistMult [150] on link prediction task over multiple datasets. Tucker decomposition for a tensor $\boldsymbol{X} \in \mathbf{R}^{I \times J \times K}$, gives tensor $\boldsymbol{Z} \in \mathbf{R}^{P \times Q \times R}$ and matrices $\boldsymbol{A} \in \mathbf{R}^{I \times P}, \boldsymbol{B} \in \mathbf{R}^{J \times Q}$, and $\boldsymbol{C} \in \mathbf{R}^{K \times R}$ as outputs.

$$\boldsymbol{X} \approx \boldsymbol{Z} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \tag{4.1}$$

$$\phi(e_s, r, e_o) = \boldsymbol{Z} \times_1 \boldsymbol{e_s} \times_2 \boldsymbol{w_r} \times_3 \boldsymbol{e_o} \tag{4.2}$$

For link prediction tasks, $\mathbf{X}$ is the adjacency tensor, consisting of the adjacency matrices for each relation in the KG, with 1 at index (s, r, o) if a relationship exists between e_s and e_o on relation r, else 0. We set $A = C$, to obtain the same entity embeddings in both the source and

object positions. The core tensor $\mathbf{Z}$ is the set of parameters that is linearly proportional to d_e and d_r, the entity embedding size and the relation embedding size respectively and $\mathbf{Z} \in \mathbf{R}^{d_e \times d_r \times d_e}$ space. Our method learns the entity embedding matrix $\mathbf{A} \in \mathbf{R}^{n_e \times d_e}$, relation embedding matrix $\mathbf{B} \in \mathbf{R}^{n_r \times d_r}$ space as shown in Equation 4.2, where $\mathbf{e}_s$, $\mathbf{e}_o \in \mathbf{R}^{n_e \times d_e}$ are the corresponding rows of A and $\mathbf{w}_r$ is the corresponding row of B. The model is trained to minimise the Bernoulli negative log-likelihood loss function which maximises the estimated probability of a relation $\phi(e_s, r, e_o)$ over known relations. This enables the function ϕ to compute a score of the edge (e_s, r, e_o).

4.2.3.2 *Positional Embeddings*

During the conversational flow, entities are mentioned sequentially. By virtue of this, entities mentioned later have a larger effect on future recommendations. Wu et al. [140] extracted a session graph to represent the sequential nature of items belonging to a user. By leveraging the properties of such graphs, they learned latent item embeddings using graph neural networks. To avoid the extraction of the session graph and increasing the complexity of the model, we use positional embeddings as described in Vaswani et al. [130]. Positional embeddings allow us to infuse sequence information into entity embeddings.

$$\mathbf{POS}_{(\mathbf{pos},2\mathbf{i}+1)} = \cos(pos/1000^{2i/d_{model}}) \tag{4.3}$$

$$\mathbf{POS}_{(\mathbf{pos},2\mathbf{i})} = \sin(pos/1000^{2i/d_{model}}) \tag{4.4}$$

The subscript **pos** in Equation 4.3 and Equation 4.4 refers to the index position of the entity in the conversation. The subscript **i** refers to the index in the vector representing the positional embedding. The final entity representation of the entity $\mathbf{e_i}$ seen at position **pos**, given the scaling factor β, is given by:

$$\mathbf{e}_{\mathbf{i},\mathbf{pos}} = \mathbf{e_i} + \mathbf{POS}_{\mathbf{pos}}/\beta \tag{4.5}$$

4.2.4 User Representation

A user is represented by a set of entities already encountered in the conversation. If a user has interacted with **n** entities, the user representation is represented by the matrix $\mathbf{U} \in \mathbf{R}^{\mathbf{nxd}}$. We then map the matrix **U** to a vector in $\mathbf{R}^{\mathbf{d}}$ space so that we can find the similarity of the user with different entities. We borrow a part of the soft-attention used by Wu et al. [140] to extract a representation of the user in $\mathbf{R}^{\mathbf{d}}$ space.

$$\alpha_i = \mathbf{q}^\top \sigma(\mathbf{W_1 U_n} + \mathbf{W_2 U_i} + \mathbf{c}) \tag{4.6}$$

$$\mathbf{u} = \sum_{i=1}^{n} \alpha_i \mathbf{U_i} \tag{4.7}$$

Here, $\mathbf{U_i}$ is the representation of the user when $\mathbf{i}$ entities have been encountered. $\mathbf{q} \in \mathbf{R^d}$ and $\mathbf{W_1}, \mathbf{W_2} \in \mathbf{R^{dxd}}$ are trainable parameters.

4.2.5 Similarity Score of Users with Movies

The final similarity scores of users with the movie entities are calculated by computing the dot product of the user representation with the movie entity representations passed through a softmax layer.

$$\mathbf{P_{rec}} = softmax(\mathbf{u} \bigodot \mathbf{E^T}) \tag{4.8}$$

We optimise our model using the cross-entropy loss on the set of recommended movies.

4.3 EXPERIMENTAL SETUP

This section presents the parameters to train our system and the evaluation metrics used to evaluate our approach. Furthermore, we describe previous approaches, against which we compare our proposed approach.[1]

4.3.1 Dataset

For this section of experiments, we use the ReDial dataset [64], which is an annotated set of dialogues where a recommender system suggests movies to a recommendee. The dataset consists of 10,006 dialogues with a total of 182,150 utterances. Following Chen et al. [18], we split the dataset into 80% for training and 10% each for validation and testing.

As suggested by Chen et al. [18], we use DBpedia as the KG for this task. It is an open KG that stores structured content from Wikimedia projects. As DBpedia is served as open linked data, as suggested by the KBRD task [18], entities (movies, actors, directors, etc.) present in the conversations are linked to DBpedia entities using the method put forward by Daiber et al. [19], where the authors perform entity linking using a generative probabilistic model. Entities are furthermore split into two categories: movie entities and mentioned entities. Movie entities are those entities, which are annotated movies present in the ReDial dataset. Mentioned entities are the entities that are linked to DBpedia as suggested in Chen et al. [18]. Entities related to actors, producers, and directors are instances of mentioned entities. For example, in Figure 4.1, "*A Nightmare on Elm Street*" and "*Happy Death Day*" are examples of movie entities, while "*Jason Blum*" is an example of a mentioned entity. The selection of our datasets allows us to investigate the knowledge retrieval from a KG by a chatbot based on the given context. The dataset includes 6,924 movie entities and 4,765 mentioned entities. Among the movie entities, 824 movies from the dataset could not be linked to DBpedia. Similarly, 121 mentioned entities could not be linked to DBpedia. We introduce the unlinked entities as isolated nodes in the graph. Since the total number of nodes and edges present in the DBpedia KG has a magnitude of 10^6 and

1 The code and the models are available at https://github.com/rajbsk/KG-conv-rec

10^7 respectively, we need to extract a relevant subgraph for this task. We build the initial set of subgraphs by considering N-hop neighbours of the movie nodes present in our dataset. When extracting a graph using the N-hop technique, the subgraph grows exponentially which induces noise in the subgraph. To counteract this issue, we use the PageRank algorithm [91] to extract subgraphs for the task.

4.3.2 Baseline Methodologies

We consider KBRD [18] as the baseline for our work. The KBRD system extracts different movie entities and mentioned entities from the dialogues and links the entities to DBpedia using the entity linking technique described by Daiber et al. [19]. The system then extracts a subgraph by considering 2-hop neighbours of movie entities. The entity embeddings are learned by using Relational Graph Convolutional Network (RGCN) [109]. The entity embeddings belonging to a particular user are then passed to a self-attention layer to learn the user representation. The dot-product of the user representation is computed with the entity embeddings and the scores are passed through a softmax layer to get the recommendation probabilities. Our proposed methodology distinguishes itself from the KBRD system by utilising different subgraphs for movie entity recommendation to users. Moreover, our approach leverages Tucker embeddings instead of RGCN embeddings, enhancing the learning of embeddings for isolated nodes in the graph. Additionally, we incorporate positional embeddings into the entity embedding, enabling the model to capture the sequential occurrences of entities in the conversation. We also compare the performance of our model with ReDial [64]. ReDial is a CRS with a dialogue generation component built using HRED [116]. The recommendation system makes use of an autoencoder model and sentiment analysis model to suggest movies to users.

4.3.3 Evaluation Metrics

We evaluate the different conversational recommendation models based on the following:

- *Mean Reciprocal Rank (MRR)* – It is defined as the average harmonic mean (across all turns) based on the rank of the first correct response entity reported, given the dialogue history. Mathematically, $\text{MRR} = \frac{1}{|D_t|}\sum_{i=1}^{|D_t|}\frac{1}{rank_{e_i}}$, where $rank_{e_i}$ denotes the rank of the first correct response entity returned for dialogue turn D_{t_i}, and $|D_t|$ denotes the total number of dialogue turns.

- *Recall@k (R@k)* – It measures the fraction of the total number of correct response entities (within the response text) retrieved in the top-k entities by the methodologies. We report for $k = \{1, 10, 50\}$. Mathematically, we have,

$$\text{R@k} = \frac{\text{\# correct response entities retrieved}}{\min(\text{\# total response entities}, k)} \tag{4.9}$$

Table 4.3: Performance of models on recall@1, recall@10, recall@50 and MRR. The confidence intervals were calculated by conducting Student's t-test on 24 runs of each of the models ($p \ll 0.001$).

Model	R@1	R@10	R@50	MRR
Li et al. [64]	2.23±0.24	13.14±0.63	29.22±0.77	0.060±0.0018
Chen et al. [18]	3.00±0.20	16.30±0.30	33.80±0.70	0.066±0.0019
2-hops	3.26±0.13	17.15±0.52	34.35±0.58	0.078±0.0014
3-hops	**3.39**±0.15	18.00±0.40	35.33±0.35	0.082±0.0018
5-hops	3.36±0.14	**18.05**±0.45	**35.70**±0.37	**0.082**±0.0016
PageRank	3.28±0.14	17.53±0.46	35.20±0.39	0.080±0.0019
PPR($\alpha = 0.9$)	3.29±0.16	17.99±0.45	35.24±0.43	0.081±0.0016
PPR($\alpha = 0.7$)	3.31±0.17	17.21±0.43	34.28±0.48	0.079±0.0017

4.3.4 Implementation Details

For learning the entity embeddings using Tucker decomposition, we set the learning rate to 0.0001 and batch size to 1,536 for all of the subgraphs. We set the dropout values of all the layers to 0.1 for every layer. For the recommendation engine, we set the batch size to 32, and the learning rate to 0.001 to avoid overfitting of the model. Similar to Chen et al. [18], we use the Adam optimiser [51] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ to optimise our model. We set the value of the scaling factor β to 1,000. We conducted grid-search and set the hyperparameters to the best performing values.

4.4 RESULTS AND DISCUSSION

First, we perform a quantitative evaluation of the models. We then analyse the effect of different subgraphs on the performance. Thereafter, we proceed towards studying the performance while recommending disconnected entities and the effect of session sequence length. Finally, we analyse two examples, where the model results are not consistent with the gold standard, to get insights for future work.

4.4.1 Quantitative Evaluation

In this work, we put forward the hypothesis that the use of pre-trained embeddings of entities supplemented with positional information improves the performance of the recommendation engine. The results in Table 4.3 show the performance of different models. To compute the performance of each model, we did 24 runs of our models to obtain the distribution of the performance scores on a different metric. We compute the confidence interval for the performance of each model on a different metric by using the Student's-t distribution [119]. Two sample t-test showed that our results are statistically significant results to the previous models ($p \ll 0.001$). We achieve a performance of 3.36%, 18.05%, 35.70%, and 0.082 on the recall@1, recall@10, recall@50, and MRR metrics respectively with 5-hops subgraph. We obtain relative improvements over the previous baseline [18] by 12.00%, 10.73%, 5.32%, and 24.24% on the four metrics respectively. These results show the efficacy of our choice of subgraphs and embeddings.

Table 4.4: Evaluation on disconnected entities. We report the performance values on different metrics for the instances when the target entity is disconnected from the already seen set of entities in the subgraphs.

Subgraph	Recall@1	Recall@10	Recall@50	MRR
Chen et al. [18]	1.41	12.01	34.28	0.051
2-hops	**2.23**	**12.84**	**36.75**	**0.060**

4.4.2 Performance Analysis

In this section, we conduct a study on the performance of the model in the presence of different subgraphs. Additionally, we analyse the recommendation performance for isolated entities in the KG. We also evaluate the performance of the model with varying session lengths. Finally, we study a few example cases wherein the results of our proposed methodology are different from the ground-truth recommendations.

4.4.2.1 Effect of different subgraphs

Graphs with a higher number of nodes and edges contain more information. However, in the recommendation scenario, it is desirable to have subgraphs from KGs containing the maximum amount of information with a minimal amount of noise. Table 4.3 shows that for models with subgraphs extracted using the N-hop method, as the value of N increases, the performance of the model improves. However, the model performance when N=5 is not significantly better than the case when N=3. This points us towards the direction of analysing the performance of the models when using subgraphs extracted using PageRank. The results show that even though subgraphs extracted using PageRank or personalised PageRank contain a lower number of nodes and edges (Table 4.1), they perform at par with N-hop models. Since PageRank identifies important nodes in a graph, PageRank graphs are subgraphs of 5-hops graphs containing a lower number of nodes and edges as well as a lesser amount of noise. Table 4.3 demonstrates that the incorporation of a weighting scheme that prioritises movie nodes in the extraction of PageRank subgraphs resulted in improved performance. Specifically, this weighting scheme enabled the PageRank algorithm to focus on the movie nodes, resulting in a subgraph that emphasises these nodes. Therefore, this methodology can be used for effective domain adaptation. When pre-training entity embeddings using the method described by Balazevic et al. [10], the execution time is directly proportional to the number of entities and relations. With a smaller and richer subgraph, the pre-training is faster while providing comparable performance.

4.4.2.2 Recommending Disconnected Entities

Disconnected target entities are those entities that are not connected through a path of any length in the KG to entities that have already interacted with the user. We compare the 2-hop model against KBRD. The two models make use of the same DBpedia subgraph and thereby have the same set of disconnected entities. We do not compare KBRD against our best-performing model since the subgraph would change for the 5-hop model. We do not compare against ReDial as it uses auxiliary information such as sentiment information which neither KBRD nor our model uses. While ReDial uses auxiliary information, neither KBRD nor our model leverages

Figure 4.3: Performance of different models on Recall@50 with the number of items. As the number of items increases, our models have better performance compared to the baseline. Also, as the number of nodes and edges increase in N-hop, the performance improves. PageRank subgraph models perform similarly when the number of items increases.

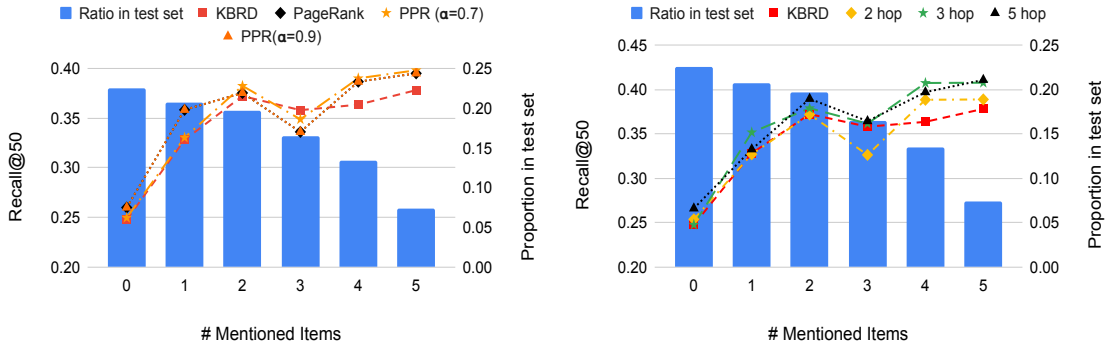

Table 4.5: Examples where our model produces inconsistent results. We take 3 cases from the test set to study the errors. The recommended movie in Dialogue 1 occurs in the fourth position and not in the first. For Dialogues 2 and 3, our model gives out the same set of recommendations for the first four movies even though the context is different.

Dialogue 1	Dialogue 2	Dialogue 3
Rec: Hello there! User: Hi. Rec: Tell me what kind of movies would you like to watch? User: I like all types of movies. Especially comedy and family Rec: I can recommend you Get Out	User: I would like to watch any movie. User: Tell me any movie. User: Like Avengers: Infinity War. Rec: Have you seen Scary Movie?	Rec: Tell me what would you like to watch. Rec: I just watched Avengers: Infinity War Rec: Have you seen it? Rec: or do you like Scary movies? Rec: have you ever seen Click? User: I like a lot of different movies. Thinking about some comedies something like Billy Madison Rec: have you ever seen You Don't Mess with the Zohan?

such information apart from the information through KGs, our models can be adapted into the domain more quickly as compared to ReDial. The ReDial model does not make use of KGs, hence there is no scenario of disconnected entities.

Table 4.4 shows the comparison of our model against KBRD. Our model performs better than KBRD in all four metrics. The results show the correctness of our assumption that using pre-trained embeddings is more helpful in improving the performance of recommendation models using KGs.

4.4.2.3 Impact of Session Sequence Length

We analyse the performance of our model on different subgraphs against the baseline. We compare the performance of different models on recall@50 which is shown in Figure 4.3. The figures show that as the number of mentioned items increase, the performance of the model improves. This is attributed to the fact that when the number of mentioned items increases, the model has more information about the user affinities and thereby captures the user preferences better. Figure 4.3 also shows that our model on different subgraphs outperforms the baseline. Also, embeddings learned using graphs with a higher number of entities and relationships have better performance. This can be attributed to the fact that a higher number of entities and relations help in learning more information and thereby better entity representations.

Target Movie	1st	2nd	3rd	4th
Get Out	Black Panther	Jumanji	It	Get Out
Scary Movie	Avengers: Infinity War	Black Panther	Thor: Ragnarok	Wonder Woman
You don't mess with the Zohan	Avengers: Infinity War	Black Panther	Thor: Ragnarok	Wonder Woman

4.4.2.4 *Error Analysis*

Table 4.5 displays three dialogues where our model does not perform correctly. In the case of Dialogue 1, the correct target movie is *Get Out*. However, our model assigns it the fourth highest probability. This can be explained by the fact that when no initial entities are mentioned in the dialogue, the model is not able to infer the context of the conversation. As of result of this, the model gives the same output irrespective of the context.

For Dialogues 2 and 3, the model gives the same result for both of the dialogues even though the context and the mentioned entities are different. Both dialogues have movies in common and the results are biased towards that movie. The model does not discriminate the movies present in the user utterance or the recommendation utterance.

4.5 SUMMARY

In this chapter, we study the impact of the size and vocabulary of KGs on the performance of the *structured knowledge selector* component of chatbots in a conversational recommendation setting. We introduce different subgraphs for the task to show the need for the extraction of appropriate subgraphs from KGs to propagate relevant information into the structured knowledge retrieval modules of a chatbot. Additionally, we propose a novel neural network architecture leveraging entity recency information as our knowledge retrieval component. We show that our model performs better than current-state-of-the-art on multiple metrics by considering the use of pre-trained KG embeddings and positional embeddings. We showcase that our model performs better while recommending disconnected entities in KG. Our investigations revealed that as the number of entities mentioned in the text increases, the performance of our model improves due to the better capture of the user profile using the historic entities, thus mitigating challenges in capturing user preferences.

5 CONVERSATION EXPLANATION USING KNOWLEDGE GRAPHS

In this chapter, we tackle the challenge of explainability in chatbots and study the explainability component in Figure 1.2. Knowledge-grounded chatbot systems utilise external knowledge, such as KGs, to generate informative and appropriate responses. When a chatbot utilises knowledge for response generation, it becomes increasingly difficult to understand how knowledge was selected and utilised during response generation. A crucial challenge of such systems is to select knowledge from a KG pertinent to the conversation context for response generation. Even though researchers have developed models for fact selection, the explicability of such systems is limited. This limits the explanation of why a certain knowledge is generated in the response text. KGs can be a powerful tool in alleviating this issue. As KGs provide a rich graphical representation of knowledge and relationships between entities, this fact selection can be formulated as path traversal over a KG conditioned on the conversation context. By tracing a path over the KG, we can explain the conversation with the chatbot. Such paths can originate from facts mentioned in the conversation history and terminate at the facts to be mentioned in the response. These walks, in turn, provide an explanation of the flow of the conversation. This chapter proposes KG-CRuSE, a simple yet effective LSTM-based decoder that uses semantic information in the conversation history and the KG elements to generate such paths for an effective conversation explanation.

5.1 INTRODUCTION

Inducing factual information during response generation has garnered much attention in conversation systems research [84, 141, 172]. While language models [166, 169] have been shown to generate responses akin to the conversation history, they seldom contain factual information, leading to a bland conversation with the agent. Knowledge-grounded conversation systems focus on leveraging external knowledge to generate coherent responses. KGs are a rich source of factual information and can be combined with an utterance generator for a natural and informative conversational flow.

Zhou et al. [171] showed that utilising KGs in conversation systems improves the appropriateness and informativeness of the conversation. Augmenting utterances in a conversation with the KG information guides the conversational agent to include relevant entities and facts in the response. For example, Figure 5.1 shows an example conversation where a user is interacting with a conversation agent about movies. The agent has access to a KG that aids in suggesting relevant facts during the conversation flow. When responding to utterance 3, the agent can utilise information from the KG and produce relevant facts about "*Christopher Nolan*". This information would be more engaging than responding with information about "*Batman*" or "*Batman Begins*".

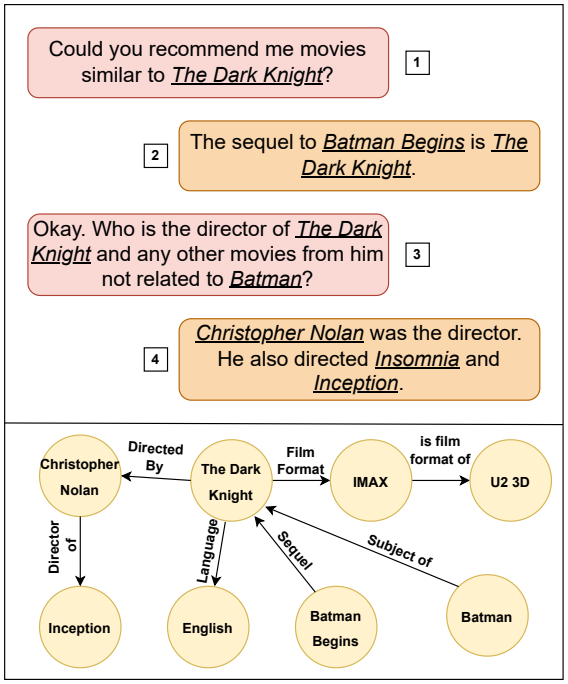

Figure 5.1: An example conversation wherein the agent utilises relevant information from the KG while generating responses. The agent generates facts about "Christopher Nolan" in utterance 4 while utilising the semantic information in the conversation history and the KG.

While KGs have been used extensively to include relevant facts in a conversation, the explicability of such systems is limited. Naturally, this fostered research on developing models for explainable conversation reasoning. Moon et al. [84] addressed this problem by inducing KG paths for conversation explainability. They posited a conversation-KG path-aligned corpus wherein utterances are augmented with a KG path to denote fact transitions in the conversation. The KG paths emanate from entities or facts mentioned in the conversation history and terminate at the entity to be mentioned in the response text. Such paths form a sequence of entities and relations and aid the conversation agent in introducing appropriate knowledge to the conversation. In addition to this, they proposed an attention-based recurrent decoder over the KG to generate entity paths. Jung et al. [46] designed a novel conversation-context infused graph neural network to propagate attention scores over the knowledge graph entities for KG path generation. While such approaches have their inherent strengths, their limitations are manifold and are detailed below.

Given a conversation context, it is desirable to generate paths that result in a natural conversation flow. Therefore it is essential to capture the semantic information in the conversation context and the KG elements. Transformer-based models [22, 56, 72] have enabled the capture of contextual relationships between different words in a sentence. Textual representations from such models have been successfully adapted for the conversation conditioned KG reasoning task [46]. However, prior works use the embedding of the [CLS] token to encode the conversation history and the KG elements. Reimers and Gurevych [103] demonstrated that such sentence embeddings are sub-optimal and lead to degraded performance in downstream application tasks and proposed sentence-transformers which are strong tools for capturing the semantic information of a sentence into a fixed-size vector. As KG elements can be long phrases, KG-CRuSE uses the SBERT model to encode both the conversation history and the KG elements for capturing their

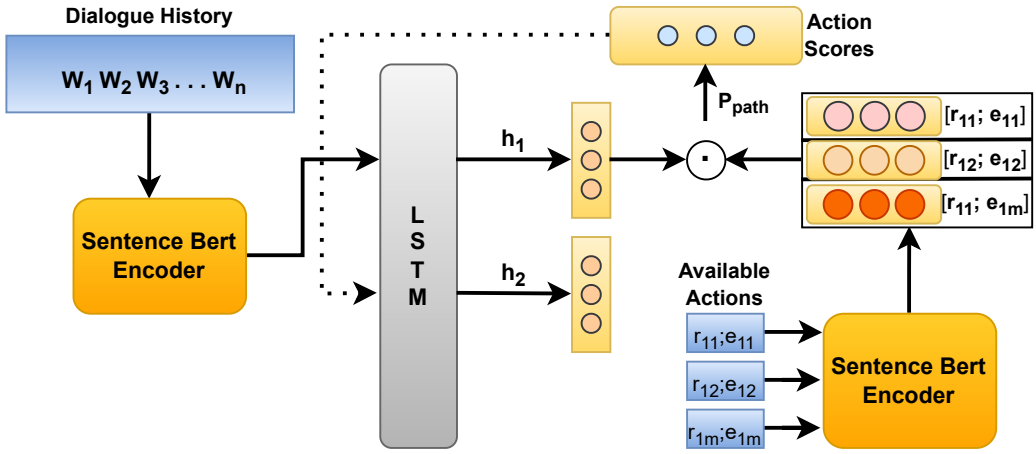

Figure 5.2: Modular overview of KG-CRuSE architecture. KG-CRuSE utilises the SBERT architecture to encode the conversation history and the KG elements. To generate walk paths over the KG, KG-CRuSE leverages an LSTM network to model the temporal information. To generate the path at timestep t, the LSTM takes as input $(\mathbf{D}, (\mathbf{r_1}; \mathbf{e_1}), ..., (\mathbf{r_{t-1}}; \mathbf{e_{t-1}}))$ and outputs the hidden state representation $\mathbf{h_t}$ of the step t. KG-CRuSE then computes dot-product of $\mathbf{h_t}$ with the embeddings of the actions available at timestep t $([\mathbf{r_{t,1}}; \mathbf{e_{t,1}}], [\mathbf{r_{t,2}}; \mathbf{e_{t,2}}], ..., [\mathbf{r_{t,m}}; \mathbf{e_{t,m}}])$ followed by a softmax layer to compute the probability of each available action.

semantic information, while making it easier to bridge the semantic gaps between the context representations and the KG representations.

As a result of the long-tailed distribution of node neighbors in a KG, it can become difficult to generate relevant paths over the KG for explainable conversation. Given the conversation history, it is desirable to traverse semantically relevant paths. KG-CRuSE utilises the rich sequential information in the conversation history and the path history to sample the top-k semantically similar neighbors for extending its walk over the KG. We show that our KG-CRuSE improves upon the current state-of-the-art on multiple metrics, demonstrating the effectiveness of KG-CRuSE for explainable conversation reasoning.

In summary, we introduce KG-CRuSE, an LSTM-based decoder that utilises SBERT embeddings to reason KG paths for explainable conversation. Our model's effectiveness is demonstrated by surpassing the current state-of-the-art performance on the OpenDialKG dataset across multiple metrics. Furthermore, we conduct an extensive empirical analysis to highlight KG-CRuSE's efficacy for the reasoning task. To promote reproducibility and future comparisons, we provide our system and baseline models as an open-source toolkit on GitHub[1].

5.2 METHODOLOGY

In the following sections, we begin with formally introducing the problem statement. We then outline the embeddings used in KG-CRuSE. Following this, we discuss the architecture of KG-CRuSE as illustrated in Figure 5.2. Finally, we describe the decoding process used by KG-CRuSE during the inference step.

1 https://github.com/rajbsk/kg-cruse

5.2.1 Formal Problem Definition

We describe the problem similar to Moon et al. [84]. The KG is defined as $\mathcal{G} = \mathcal{V}_{\mathcal{KG}} \times \mathcal{R}_{\mathcal{KG}} \times \mathcal{V}_{\mathcal{KG}}$, where $\mathcal{V}_{\mathcal{KG}}$ is set of entities and $\mathcal{R}_{\mathcal{KG}}$ is set of relations in the KG. Facts in the KG are denoted by triples, and each has the form (e, r, e') where $e, e' \in \mathcal{V}_{\mathcal{KG}}$ are entities and $r \in \mathcal{R}_{\mathcal{KG}}$ is the relation connecting them.

In addition to the KG, each input contains a conversation $D \in \mathcal{D}$, represented as a sequence of utterances $D = \{s_1, ..., s_n\}$, and the set of entities $x_e = \{x_e^{(i)}\}$ occurring in the user's last utterance s_n, where $x_e^{(i)} \in \mathcal{V}_{\mathcal{KG}}$. The output is represented as $y = \{y_e, y_r\}$, where y_e is a set of entity paths $y_e = \{y_e^{(i)}\}$, with each element $y_e^{(i)} = \{y_{e,t}^{(i)}\}_{t=1}^{T}$ denoting an entity path connecting $x_e^{(i)}$ to the response entity $y_{e,T}^{(i)}$. Likewise, $y_r = \{y_r^{(i)}\}$ is a set of relation paths, where $y_r^{(i)} \in \mathcal{R}_{\mathcal{KG}}$. The element $y_r^{(i)} = \{y_{r,t}^{(i)}\}_{t=1}^{T}$ is a sequence of relations from the KG connecting $x_e^{(i)}$ and $\{x_{e,t}^{(i)}\}_{t=1}^{T}$.

5.2.2 Conversation and KG Representation

Capturing the semantic information in the conversation context is an important component of our model. SBERT is a contextual sentence encoder that captures the semantic information of a sentence in a fixed-size vector. We encode pieces of text using Equation 5.1. The text is first sent through a pre-trained BERT model to obtain the contextual representation of its tokens. The sentence embedding is computed using a mean pool of the contextual token representations. The conversation context is constructed by concatenating a maximum of three previous utterances and is then passed through SBERT encoder to obtain a fixed-size contextual conversation representation.

$$\mathbf{S} = \text{MeanPooling}(\text{BERT}(S)) \tag{5.1}$$

To align the semantic vector space of the conversation representations and the KG representations, we use SBERT to encode the KG elements. As KG entities and relations can be words or phrases, SBERT can effectively capture their semantic information. We use the publicly available SBERT-BERT-BASE-NLI[2] model with mean-pooling as our SBERT encoder.

5.2.3 KG-CRuSE Architecture

KG-CRuSE learns to traverse a path on the KG by learning a function π_θ that calculates the probability of an action $a_t \in \mathcal{A}_t$ given the state s_t. The state s_t contains the conversation history and entities already traversed by KG-CRuSE while decoding the paths, while

macro:

macro:

$\mathcal{A}_t$ is the set of edges from the KG available to KG-CRuSE for extending its path.

The state s_t at step t is defined as a tuple $(D, (r_1, e_1, ..., r_{t-1}, e_{t-1}))$, where D is the conversation context and $r_i, e_i (i < t)$ are the relation and entity already decoded by KG-CRuSE at step i. The initial state s_0 is denoted as $(D, \varnothing)$, where $\varnothing$ is the empty set.

At step t, an action has the form $a_t = (r_t, e_t) \in \mathcal{A}_t$, where $\mathcal{A}_t$ is the set of all possible actions available to the model at step t. $\mathcal{A}_t$ includes all outgoing edges of e_{t-1} in the KG $\mathcal{G}$, i.e. $\mathcal{A}_t$ is the set of all the outgoing edges of the entity decoded by KG-CRuSE at timestep $t-1$. To

2 https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

let the agent terminate the search process, we add self-loop edges to every entity node in the graph denoting no operation ("self-loop"). The action a_t is represented as a concatenation of the relation and entity embedding $\mathbf{a}_t = [\mathbf{r}_t; \mathbf{e}_t]$, where $\mathbf{r} \in \mathbb{R}^{d_r}$, $\mathbf{e} \in \mathbb{R}^{d_e}$ and $\mathbb{R}^{d_e}$, $\mathbb{R}^{d_r}$ are the size of the entity embedding and relation embedding respectively. At step 1, KG-CRuSE chooses between the entities mentioned in s_0 for path traversal. The relation associated with action at step 1 is the zero vector. As mentioned, the state s_t contains the conversation context and action history (path history). This sequential information in s_t is modelled using an LSTM:

$$\mathbf{d} = \mathbf{W}_d \mathbf{D}_{emb} \tag{5.2}$$

$$\mathbf{h}_0 = \text{LSTM}(\mathbf{o}, \mathbf{d}) \tag{5.3}$$

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{a}_{t-1}), t > 0 \tag{5.4}$$

where $\mathbf{D}_{emb}$ is the contextual conversation embedding obtained using Equation 5.1 and $\mathbf{W}_d$ is a learnable matrix that maps the conversation embedding to the LSTM input dimension. Given the hidden state representation $\mathbf{h}_t$ at time t, KG-CRuSE assigns a probability to each action using Equation 5.6.

$$\mathbf{x}_t = \mathbf{W}_{3,\theta}(\text{ReLU}(\mathbf{W}_{2,\theta}\mathbf{h}_t^\intercal)) \tag{5.5}$$

$$\pi_\theta(a_t|s_t, \mathcal{A}_t) = \frac{\exp(\mathbf{a}_t \cdot \mathbf{x}_t)}{\sum_{a_i \in \mathcal{A}_t} \exp(\mathbf{a}_i \cdot \mathbf{x}_t)} \tag{5.6}$$

The hidden state representation $\mathbf{h}_t$ is passed through a two-layered dense network with ReLU activation [85] in the first layer. The LSTM weights, $\mathbf{W}_{2,\theta} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_{3,\theta} \in \mathbb{R}^{(d_r+d_e)\times d_h}$ are the learnable parameters, and d_h is the LSTM hidden representation size.

5.2.3.1 Model Learning

We train KG-CRuSE by minimising the cross-entropy loss on the entities decoded at each timestep. Additionally, we train the model using teacher forcing [121], wherein the model makes each action conditioned on the gold history of the target path. As fine-tuning SBERT architectures degrade the model performance as shown in Section 5.4.3, during the training process we do not fine-tune the SBERT architectures, but back-propagate the gradients to the entity and relation embeddings.

5.2.3.2 KG-CRuSE Path Generation

Once the model is trained, KG-CRuSE takes the conversation history and the entities mentioned in the current utterance as input, a horizon T as the maximum length of generated paths and outputs a set of entity paths, relations paths of length T along with the probability score of each path. During inference, we remove self-loops from the KG except for the self-loop with the label "self-loop" introduced in Section 5.2.3, denoting the termination of path exploration. We do so to allow the agent to traverse diverse paths rather than staying at entities mentioned in the conversation history.

5.3 EXPERIMENTAL SETUP

This section presents the dataset used, the baselines compared with and the description of the model settings of KG-CRuSE along with the metrics the models have been evaluated on.

5.3.1 Dataset

We evaluate our proposed framework on the OpenDialKG dataset. The dataset has 91,209 turns spread over 15,673 dialogues in either task-oriented dialogues (recommendations) or chit-chat conversations on a given topic. Each turn is annotated with a KG path to represent fact transitions in the conversation. The KG is a subset of the Freebase KG [13], which has 1,190,658 fact triples, 100,813 entities and 1,358 relations. Following Moon et al. [84], we split the dataset randomly into 70%, 15% and 15% for training, testing and validation.

5.3.2 Baselines Methodologies

We compare KG-CRuSE against the following baseline models suggested by Moon et al. [84] and Jung et al. [46]:

- Tri-LSTM [156]: The model encodes each utterance along with facts from the KG within 1-hop distance from the entities mentioned in the current utterance. This retrieves facts from the KG for conversation explanation.
- Ext-ED [95]: The response generation is conditioned on external knowledge vectors for generating response entity tokens at the final softmax layer, without using the structural information from the KG.
- Seq2Path [46]: An attention-based Seq2Seq model is modified to generate entity paths by masking out unreachable nodes at each decoding step.
- Seq2Seq: An LSTM-based seq2seq [121] model where the decoder is modified to generate entity paths. We use modality attention as the output of the encoder and replace the softmax layer in the decoder with a zero-shot learning layer in the KG embedding space.
- DKGW [84]: A model to generate KG paths using domain-agnostic, attention-based recurrent graph decoder reinforced with a zero-shot learning layer over the KG embedding space.
- AttnIO [46]: A conversation conditioned KG path traversal leveraging attention flow using graph neural networks.

Since the authors of OpenDialKG and AttnIO have not released their implementations, we report their performance on our re-implementations. We note that for most systems, our implementation is similar to or better than the reported results. Regarding AttnIO, we were not able to reproduce the original results. However, we note that errors in implementing the node sampler or leakage of the test dataset into the training dataset can easily lead to an overestimated accuracy. .

Model	Recall@k path@1	path@5	path@10	path@25	tgt@1	tgt@5	tgt@10	tgt@25
Tri-LSTM	3.2	22.6	36.3	56.2	-	-	-	-
Ext-ED	1.9	9.0	13.3	19.0	-	-	-	-
Seq2Path	14.92	31.1	38.68	48.15	15.65	33.86	42.52	53.28
Seq2Seq*	6.53±0.78	26.21±1.21	35.02±1.27	45.78±1.18	7.13±0.85	30.64±1.62	41.01±1.43	52.97±1.55
DKGW*	14.16±1.16	37.26±1.91	47.85±2.60	59.20±2.33	14.96±1.04	39.53±1.81	51.06±2.15	63.85±1.58
AttnIO*	19.08±1.19	38.49±0.79	43.99±1.10	48.94±0.55	**20.32±1.80**	45.90±0.93	52.82±0.65	55.17±0.96
KG-CRuSE	**19.59±0.43**	**44.62±1.08**	**56.16±1.21**	**70.59±0.38**	20.20±0.36	**47.76±0.62**	**60.11±0.92**	**75.30±0.57**

Table 5.1: Performance of KG-CRuSE in comparison with other baseline methods on different Recall@k metrics. The numbers reported are the mean values with the sample standard deviation (p=0.01). Results are statistically significant with p=0.01. Models with * denote our re-implementation.

5.3.3 Evaluation Metrics

We evaluate our models on different recall@k metrics for entity and path retrieval. Path@k measures if the ground-truth path is present in the top-k paths with the highest probability searched by the agent. Similarly, tgt@k measures if the response entity is present in the top-k entities retrieved by the agent. In situations where multiple paths point to the same response entity, we consider the path with the highest score for entity retrieval.

5.3.4 Implementation Details

For the task, we set horizon T to 3. The conversation, entity and relation embeddings are encoded using SBERT into a 768 dimensional vector. In KG-CRuSE, we consider 3 LSTM layers with $d_h = d_e + d_r = 1,536$, where d_e and d_r are the dimensions of the entity and relation embeddings respectively. To prevent the agent from overfitting on the dataset, we add L_2 regularisation with a weight decay parameter of 1e-3.

Similar to Jung et al. [46], we set the batch size to 8 and train the model with Adam optimiser [51] with a learning rate of 1e-4 for 20 epochs. The hyperparameters are determined using grid-search. For models with re-implementations, we report the results on five different data splits. For Tri-LSTM and Ext-ED, we report the number reported by Moon et al. [84], while for Seq2Path, the numbers are reported from the work of Jung et al. [46]. As entity occurrences in a conversation dataset are sparse, improper data splits can lead to an overestimation of the model performance due to the presence of similar entities in the training and the test set. Hence, it is desirable to report the performance on five different splits of the data rather than an assessment of five models on one split.

5.4 RESULTS AND DISCUSSION

We begin with performing a quantitative evaluation of the models. Following this, we study the impact of our choice of sentence embeddings on the model performance. Furthermore, we analyse the impact of beam width at each decoding step during inference. Finally, we provide insights into examples where the results of KG-CRuSE are not consistent with the ground truth paths.

Model	P@1	P@25	E@1	E@25	Rel@1
BERT	12.74	66.72	12.98	72.14	39.37
ALBERT	13.42	65.67	13.96	72.23	**40.93**
SRoBERTa	17.17	68.04	17.65	73.34	40.71
SBERT	**19.52**	**70.72**	**20.20**	**75.72**	40.02

Table 5.2: Influence of sentence embeddings on KG-CRuSE performance. Comparison of different embedding methods.

SBERT Fine-tuned	Aligned KG	P@1	P@25	E@1	E@25	Rel@1
Yes	No	17.82	69.47	18.21	74.47	40.24
Yes	Yes	18.46	69.93	19.00	74.75	40.47
No	No	18.00	62.01	18.52	74.54	38.48
No	Yes	**19.52**	**70.72**	**20.20**	**75.72**	**40.02**

Table 5.3: Results on fine-tuning the SBERT architecture used for encoding the conversation history.

5.4.1 Quantitative Analysis

In this section, we provide insights on the performance of our proposed approach based on the OpenDialKG dataset. From Table 5.1, it can be observed that KG-CRuSE performs better than the different baseline models. For entity and path accuracy, AttnIO has the closest performance compared to our model, with the latter being 2.7% relatively better on both path@1 and entity@1 metrics. On increasing k of recall@k, we find KG-CRuSE has at least 10% relative improvement over the baseline models. It is interesting to notice that on increasing the value of "k", KG-CRuSE performs relative better than other models. KG-CRuSE identifies paths semantically relevant to the conversation context but different from the gold-label paths as discussed in Section 5.4.6. The huge gain on the path@25 metric advocates for this hypothesis. It is worthwhile to notice that although AttnIO has the closest performance for path@1 and entity@1 compared to KG-CRuSE, the performance of path@k and entity@k does not improve fast enough when "k" increases. This might be due to the fact that the beam width reported by the authors is not expressive enough to capture semantically relevant paths or entities.

5.4.2 Effectiveness of Sentence Embeddings

In our framework, we utilise sentence SBERT embeddings to encode conversation context and KG elements. In this section, we conduct an ablation study on the efficacy of such embeddings. We replace the SBERT embeddings with the *[CLS]* token representation of BERT[3] [22] and ALBERT model[4] [56] in KG-CRuSE. Additionally, we consider an instance wherein the elements are encoded using Sentence-RoBERTa (SRoBERTa)[5] [103]. The results in Table 5.2 demonstrates the strength of our embedding choices wherein SBERT and SRoBERTa outperforms the BERT and ALBERT embeddings. Both SBERT and SRoBERTa sentence embedding models are pre-trained on NLI datasets, which allows them to capture rich semantic information for textual similarity. These embeddings have demonstrated strong performances in the task of semantic search using

3 https://huggingface.co/bert-base-uncased
4 https://huggingface.co/albert-base-v2
5 https://huggingface.co/sentence-transformers/roberta-base-nli-mean-tokens

Size	P@1	P@25	E@1	E@25
2, 5, 50	**19.59**	56.26	**20.09**	62.19
2, 10, 25	19.55	64.93	20.04	70.16
2, 10, 50	19.55	64.93	20.04	70.18
2, 25, 10	19.52	69.75	20.02	74.57
2, 25, 25	19.52	**70.72**	20.02	75.72
2, 25, 50	19.52	**70.72**	20.02	**75.75**
2, 50, 5	19.52	68.46	20.02	72.53
2, 50, 25	19.52	70.56	20.02	75.43

Table 5.4: Impact of the beam width at different timesteps on the model performance. The results are reported on one of the dataset splits. The best results are shown in bold, while the results on the default setting of KG-CRuSE are underlined. All numbers are in percentage.

cosine-similarity [103]. It should be noted that before the softmax layer in KG-CRuSE, we compute the dot product of the LSTM layer hidden representation with that of the relation-entity embeddings available at the given timestep. As a result of this step, we expect the performance of SBERT and SRoBERTa embeddings to be better than BERT and ALBERT embeddings.

Additionally, we see from Table 5.2 that the relation accuracy of all models are higher than the path accuracy. This is due to the outgoing edges of a node (from the conversation history) sharing similar features if they are connected using the same relation. Thus multiple entities can fit our choice of the response entity given the conversation context.

5.4.3 Impact of KG Embedding Alignment and SBERT Fine-tuning

In this section, we study the impact of encoding KG elements with SBERT embeddings. Additionally, we analyse if fine-tuning the SBERT architecture used for encoding the conversation history is beneficial for explainability.

Table 5.3 outlines four situations, where in two cases, we fine-tune the SBERT architecture used for encoding the conversation history. We also consider two cases where the embeddings of the KG elements are initialised with values drawn from a normal distribution with mean 0 and standard deviation 1, corresponding to the value "No" in the second column. It should be noted that we never consider fine-tuning the SBERT architecture used for encoding the KG elements as that would require a large memory requirement that is proportional to the number of actions available to the agent..

We see from the Table 5.3 that in cases when the KG elements are not encoded with SBERT embeddings corresponding to the value of *No* for *Aligned KG*, the performance drops as compared to cases when we use SBERT embeddings. Additionally, we find that fine-tuning SBERT leads to a decrease in the performance of KG-CRuSE. This can be attributed to the change in semantic space of the conversation embeddings and the KG embeddings during fine-tuning. Hence, we do not finetune the SBERT architecture in the default setting of KG-CRuSE.

5.4.4 Impact of Beam-Width on Path Reasoning

In this experiment, we study the influence of beam width at different timesteps on the model performance. The first column of Table 5.4 lists the tuples (K_1, K_2, K_3) where each K_i denotes

Model	GPU	Train Time	Test Time
Seq2Seq	Nvidia 1080Ti	≈8 mins	≈1 mins
DKGW	Nvidia 1080Ti	≈4 mins	≈8 mins
AttnIO	Tesla V100	≈38 mins	≈82 mins
KG-CRuSE	Nvidia 1080Ti	≈7 mins	≈8 mins

Table 5.5: Analysis of the time required by different models for training and inference on the OpenDialKG dataset. The numbers in the third column denote per epoch train time.

Conversation	Model	Walk Path
A: Could you recommend movies similar to Kung Fu Panda? B: [response]	KG-CRuSE Ground Truth	Kung Fu Panda→written by→Cyrus Voris Kung Fu Panda→directed by→Mark Osborne→wrote→Monsters vs. Aliens
A: Oh cool, I also read Wocket in my Pocket! But sure, what else is there? B: Cool! Yertle the Turtle and Horton Hears a Who! are also written by Dr. Seuss. A: That first one is really old right? I think it was released in 1958. // B: [response]	KG-CRuSE Ground Truth	1958→released year→Tom's Midnight Garden Garden→has genre→Children's literature 1958→released year→Have Space Suit - Will Travel→written by →Robert A. Heinlein
A:Could you recommend a book by Jeffrey Zaslow? B: [response]	KG-CRuSE Ground Truth	Jeffrey Zaslow→wrote→The Last Lecture Jeffrey Zaslow→wrote→Last Lecture →has genre→Non-fiction

Table 5.6: Examples where KG-CRuSE generates path different from the true paths.

the top-K_i edges sampled at timestep i. The value of i was set to 3 based on the maximum path length observed in the OpenDialKG dataset.

We conduct this analysis on a single split of the dataset keeping all other parameters of the model constant. We consider a diverse set of values for each K_i. From Table 5.4, we find that although the tuples (2, 5, 50), (2, 10, 25), (2, 25, 10) and (2, 50, 5) have an equal number of sampled paths, tuple (2, 25, 10) performs better than others. Interestingly, we observe that the sampling sizes at the second timestep play a significant role in finding optimal paths for KG-CRuSE. The first two sets of fact selection (i.e. during timesteps 1 and 2) largely determine the facts reachable by KG-CRuSE. Sampling more samples during the initial timesteps enable the agent to explore diverse paths initially and KG-CRuSE then makes an optimal selection of facts dependent on the conversation information.

5.4.5 Analysis of Computational Requirements

In this study, we conduct an analysis of the time required for training the model. We also compare the performance of different architectures with regard to the inference speed.

Table 5.5 shows that while DKGW has a better train time per epoch than KG-CRuSE and Seq2Seq has a better inference speed than KG-CRuSE, we can observe from Table 5.1 that our model achieves better performance compared to these models. It is worthwhile to mention that while AttnIO achieves the closest performance to KG-CRuSE as shown in Table 5.1, it requires roughly six times more training time and is ten times slower during inference. This indicates the benefits of using KG-CRuSE for explainable conversation using KGs.

5.4.6 Qualitative Analysis

This section highlights three scenarios showcasing the underlying working of KG-CRuSE. Table 5.6 displays three examples where KG-CRuSE generates paths different from the gold KG paths. In the first example, it can be observed that KG-CRuSE identifies a path that is insufficient to answer the given question. This can be due to the limited conversation context provided. Choosing this fact might lead to a conversation with the agent, however, the user query is not answered with the path chosen by KG-CRuSE.

In the second example, the relation traversed by KG-CRuSE is correct. However, as the conversation context is not specific, it decodes a path that might potentially fit the conversation context but is different from the gold path. However, in the third example, even with limited context, KG-CRuSE identifies a path relevant to the context, however, the final entity differs from the gold path. Such paths are admissible as all of them fit the conversation context appropriately.

5.5 SUMMARY

In this chapter, we propose KG-CRuSE, an LSTM-based lightweight framework for explainable conversational reasoning. We utilise SBERT embeddings to capture the rich semantic information in the conversation history and the KG elements. We conduct an extensive evaluation to demonstrate that our framework outperforms several baseline models on both explainability and response entity retrieval. Such explainable components in chatbots enhance user experiences by providing clear and comprehensible reasons for the chatbot's actions. This in turn increases trust and transparency by allowing users to see how the chatbot arrived at the response. The KG-CRuSE model relies on annotated datasets for training, which restricts its applicability in scenarios where conversations lack annotated KG paths.

6 KNOWLEDGE SELECTION FROM UNSTRUCTURED SOURCES IN CHATBOTS: EXTRACTIVE QUESTION ANSWERING

This chapter delves into the use of unstructured sources of knowledge in chatbots, emphasising their potential to expand the range of user queries a chatbot can address. Although not as systematically organised as structured sources, unstructured sources offer a broader pool of information, which can be effectively harnessed to train the QA aspect of chatbots. In a chatbot's QA module, unstructured knowledge sources are typically utilised to retrieve relevant text spans or sentences from a document or article that can potentially answer the user's query. In this scenario, the chatbot has to select knowledge from unstructured sources. Researchers have primarily focused on two related components of QA systems: *AS2* and *EQA*. Given a user question and a relevant document, AS2 systems focus on selecting candidate sentence(s) (i.e., context), from the document, that contains the probable answer (with a high probability) to the user's question. On the other hand, EQA involves the extraction of text spans from the *selected context* that correctly answers the question – also referred to in the literature as MRC. It is essential to notice that these Chapter 4 and Chapter 6 are complementary. In fact, the combination of structured and unstructured knowledge sources can be used to create highly effective chatbots [172]. By utilising structured sources to provide precise answers to specific questions and unstructured sources to understand a broader range of queries, chatbots can provide a more comprehensive and satisfying user experience. It is important to notice that gathering domain-specific training datasets is expensive and time-consuming to construct. Specifically, we train the extractive question-answering and the answer sentence selection components of a chatbot using synthetically generated datasets, in turn removing the need for obtaining manually annotated datasets. This chapter focuses on training domain-specific extractive question-answering components using synthetic datasets, while Chapter 7 focuses on training answer sentence selection modules using synthetic datasets.

QA has become a foundational research area in NLP with widespread applications in search, personal digital assistance, and conversational systems. Despite the success in open-domain question answering, existing EQA models pre-trained using Wikipedia articles (e.g., SQuAD data) perform rather poorly in *closed-domain* and industrial scenarios. Further, a major limitation in adapting question-answering systems to such contexts is the poor availability and the expensive annotation of domain-specific data. Thus, the wide applicability of QA models as standalone applications or downstream chatbot applications are severely hampered in enterprise systems. In this chapter, we aim to overcome the aforementioned issues by developing *QASAR*, a novel Extractive Question Answering (EQA) framework employing *self-supervised learning* for effective domain adaptation. For the first time, we demonstrate the benefit of fine-tuning pre-trained EQA models for closed domains using synthetically generated domain-specific questions and answers

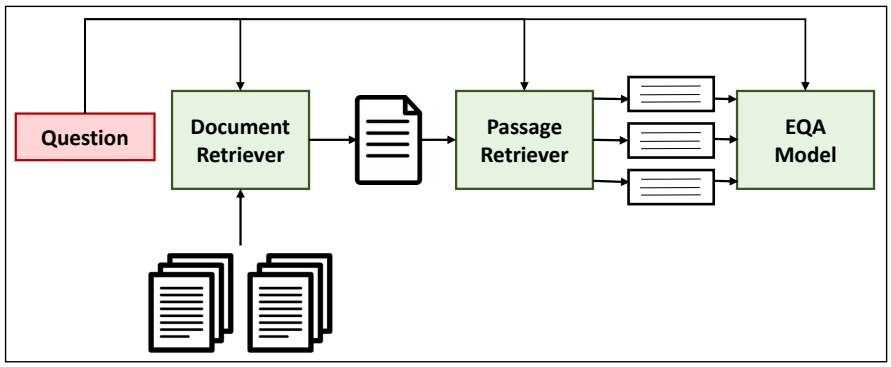

Figure 6.1: Illustrative Components of Question Answering (QA) System.

(from relevant articles) from large language models such as *T5*. In addition, we put forward a new *context retrieval* component based on question-context semantic relatedness for improving the performance of the *QASAR* EQA framework on both, open and closed domains. The experimental results demonstrate statistically significant performance increases on both open-domain and closed-domain QA datasets with minimal labelling work, which we hope will help to the ease of adoption of such systems in enterprise settings.

6.1 INTRODUCTION

NLP has emerged as a key paradigm in enterprise settings for a wide range of industrial applications such as conversational bots, question-answering systems, and intelligent search. The rapid advancement of deep learning techniques has sparked intense research interest in teaching machines to read, comprehend, and grasp organised and unstructured human texts and utterances – referred to as *Machine Reading Comprehension* (MRC) [159, 164].

EQA systems are closely connected to MRC [159] systems in that they try to automatically extract relevant answers to user queries articulated in natural free-flowing language from knowledge sources such as documents, articles, or manuals. For example, if a user queries *"When was Abraham Lincoln assassinated?"*, the question answering system is required to respond with *"Apr 15, 1865"* as the answer. However, despite its apparent simplicity, this fairly easy activity requires complex language, context, and semantic comprehension for question analysis, paraphrase identification, information retrieval, and exact contiguous answer border extraction.

EQA systems typically comprise *three* components[1], as shown in Figure 6.1: (i) *Document Retriever* is responsible for selecting the document that potentially contains the answer to the user question; (ii) *Passage Identifier* for dividing the document into contextually related passages and selecting candidate passages; and (iii) *Question Answering Extractor* which takes the question and a candidate passage as input, to extract the precise answer span. QA systems returning contiguous text spans from passages as user answers are referred to as *Extractive Question Answering* (EQA)[2]. In the remainder of this chapter, we use the term QA to denote EQA, unless stated otherwise.

1 www.section.io/engineering-education/question-answering/
2 Other typical forms of QA systems are *Cloze-style*, *Multi-choice*, and *Free-form* [164].

QA systems are common in search applications because, rather than presenting entire documents for the user to find the appropriate answer within, a QA system can efficiently extract relevant information from the documents and present it directly to the user, as seen in popular search engines such as Google Search and Microsoft Bing.

Integrating QA systems into chatbots [110] has recently attracted significant industrial attention, primarily for two reasons: For starters, it offers straightforward and automatic assistance solutions to user inquiries about the products and services offered. Second, it avoids the costly requirement for enterprises to develop structured labelled data (such as FAQs or intent categorization) for the purpose of building chatbots [7].

QA systems for specialised domains like as medical [132] and legal [44] are an active topic of research for providing preliminary information to domain specialists from a large document corpus. Indeed, QA systems (such as Google DialogFlow) are now available as a service for the seamless development of enterprise chatbots. This simplifies the process for the end user/customer, as the user uploads relevant articles (without annotation) that serve as a knowledge base for the chatbot to answer user queries.

Significant progress has been made in QA, which is one of the basic problems in NLP domain. Pre-trained transformer-based language models, such as BERT [22], are initially trained on enormous amounts of unlabeled data from common crawl corpora, and Wikipedia dumps. With the development of large labelled *open-domain* QA datasets such as WikiQA [152], SQUAD [102], and Natural Questions [55], QA systems based on subsequent *fine-tuning* of language models on such datasets have been shown to perform significantly well [28].

However, the major challenges in the wide usage of QA platform in the industry and in different fields of study are:

- *Domain Adaptation* [60] – When applied directly on "generic" open-domain data, such pre-trained QA models perform poorly when applied to *relatively limited or closed domains*, i.e., specialist fields like legal, bio-medical, and science. The difference in terminology, lexicon, and linguistic characteristics between the closed-domain and open-domain QA training data is specifically accountable for the performance decrease.

- *Domain-specific Training* [83] – Furthermore, building domain-specific labelled QA datasets for QA model training or fine-tuning is expensive, as it necessitates document chunking into a series of paragraphs (contexts) and the creation of associated question-answer pairs per context. Furthermore, the lack of domain-specific knowledge resources, such as ontologies, limits training options.

- *Non-Factoid* [158] – In general, for closed domains, question and answer texts do not overlap substantially as compared to open-domain scenarios, as the answer typically fills in missing information. Hence, semantic similarities between such domain-specific non-factoid QA could have a large gap, posing a problem to pre-trained open-domain QA models.

The above issues have severely hampered the adoption of QA systems in a broad range of specialised applications, and in industrial dialogue systems and chatbot builders such as Google DialogFlow[3].

Despite the difficulties in adapting QA for restricted domains, major efforts have been made within the industry to integrate QA models in production environments for conversation systems due to the benefits they can provide to clients. For example, Google integrated a beta

3 https://cloud.google.com/dialogflow

version of the QA platform known as *knowledge base*[4], in which a collection of relevant documents can be provided to DialogFlow (Google's chatbot builder solution) for use as domain knowledge by the chatbot to answer questions. Similarly, Amazon recommends *kendra*[5], an intelligent search engine, as part of empowering customer chatbots and agent assistants. At the same time, Microsoft introduced *QnA Maker* [6], which allows customers to build chatbots using purely unstructured text from uploaded documents. However, there is a substantial difference in QA system performance between closed-domain and open-domain applications.

We present *QASAR*, a novel *question answering framework utilising self-supervised learning* for domain adaption of language model-based EQA systems that do not require any labelled data. We first train an answer extraction and question generation model on a "generic" EQA dataset like SQuAD, and then utilise it (without additional fine-tuning) to produce synthetic question-answer pairs on specific *closed-domain* datasets (acting as synthetically generated training data). This is used for fine-tuning a pre-trained QA language model. To our knowledge, this is the first work to provide a self-supervised learning language model-based question answering. Experiment results show that our system considerably improves the performance for both open- and closed-domain datasets. Furthermore, as part of our proposed QA system, we provide a unique context retrieval mechanism based on sentence embeddings that outperforms other existing passage retrieval techniques and industrial solutions.

In summary, our contributions in this work can be summarised as follows:

- A novel *self-supervised* QA framework enabling the adoption of QA platforms across diverse open and closed application domains, without the need of any expensive training data;
- Semantic similarity-based context retrieval technique providing rich information to QA model for improved answer extraction; and,
- Extensive experiments on both open- and closed-domain datasets showcasing significant gains over existing solutions.

6.2 METHODOLOGY

In this section, we first define the problem statement and then introduce our proposed *Question Answering via Self Learning* (QASAR) framework as portrayed in Figure 6.2. It consists of *three* phases which we discuss next and introduce the implementation details of the different models composing the phases.

As seen in Figure 6.2, *QASAR* operates in 3 phases, namely:

- *Offline Training* – involves pre-training of the self-supervised modules on open-source datasets;
- *Online Fine-Tuning* – performs fine-tuning for domain adaptation of the language model-based QA framework using domain-specific documents; and,
- *Inference Time* – extracts answers to user queries from the documents.

We next describe in detail the internal modules and the working of the individual phases.

4 cloud.google.com/dialogflow/es/docs/how/knowledge-bases
5 https://aws.amazon.com/kendra/
6 https://www.qnamaker.ai/

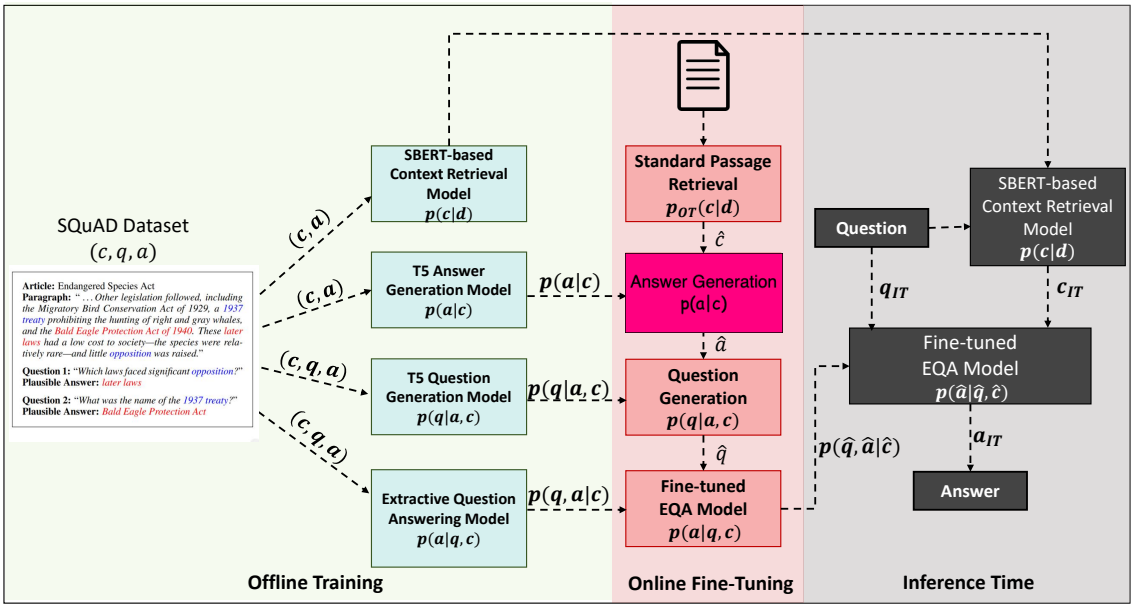

Figure 6.2: Proposed *QASAR* Framework with three phases: (a) Offline Training; (b) Online Fine-Tuning; and (c) Inference Time.

6.2.1 Formal Problem Definition

Question Answering is often formulated as a supervised learning task — given a collection of training instances, the goal of $\langle c_i, q_i, a_i \rangle$ (where c_i is a passage or context and q_i is a question) is to train a predictor for extracting the answer a_i for a question q_i from the passage c_i.

This section presents an efficient *self-supervised* EQA methodology that does not require an expensive labelled domain-specific training dataset. As a result, our problem statement includes the following:
Given a collection of closed-domain document(s), $\mathcal{D}$ (knowledge repository) and a user query q, extract the pertinent answer a (for q) from the document corpus $\mathcal{D}$.

6.2.2 *QASAR* Offline Training

The *offline training phase* is in charge of training the *QASAR* framework's *self-supervision*, *context retrieval*, and *EQA* modules using a generic dataset. In this phase, we use the SQuAD open-domain QA training data, which is one of the most well-known datasets in the literature with a relatively large number of training samples for adequate model training. *QASAR* trains four distinct models (refer Figure 6.2) in the offline stage, given a context c, question q, and response a from SQuAD:

- **Answer Generator** – Given only the context c, this model is trained to generate the associated answer a. Mathematically, this module learns the probability of occurrence of a given c, i.e., $\Pr(a|c)$.

- **Question Generator** – This model is trained to generate synthetic questions (for simulating and approximating user queries), when provided with the context and the answer. That is, it learns $\Pr(q|a, c)$.

The above two models form the backbone for *self-supervision* in *QASAR*, as they are used to generate synthetic domain-specific training data during the next *online fine-tuning* phase. For the generators, we utilise *T5-small* language model pre-trained on SQuAD, although other available models can be directly used in our framework.

- **Extractive Question Answering** – A language model-based EQA architecture is trained to extract the correct answer spans from the given context along with the question (i.e., $Pr(a|q,c)$) using the SQuAD data. Specifically, *QASAR* uses state-of-the-art SQuAD pre-trained *SpanBERT* [45] EQA model.
- **SBERT-based Context Retriever** – Uses semantic understanding between user question q and context c to learn to identify candidate sub-contexts with the highest probability of containing the answer a to the user question. We use sentence transformer SBERT-based sentence embeddings to gauge the semantic similarity between question and context to learn to select candidate sub-context for EQA.

We discuss the exact training method of the individual models in *QASAR* in Sec. 6.2.5. Take note that this offline phase is only conducted once. In the following phase, the first three trained models are used, while the context retriever is used directly in the final inference phase.

6.2.3 *QASAR* Online Fine-Tuning

The online fine-tuning process adapts the *QASAR* framework to the required domain. The goal is to create *context-question-answer* triples (c, q, a) from a set of domain-specific documents d. This is accomplished by generating the set of triples using the pre-trained self-learning models from Sec. 6.2.2, which serves as the synthetic training dataset for fine-tuning the pre-trained EQA model. The EQA model is thus fine-tuned to the domain of the provided document(s) d.

The procedure begins by separating the document d into contexts $\hat{c}$, with each context $\hat{c}_j$ containing a set number of l sentences derived by dividing the document into n paragraphs. These contexts $\hat{c}_j$, where $j \in [1, \lceil n/l \rceil]$, are then fed into the pre-trained Answer Extraction model, which generates responses $\hat{a}_i$, where $i \in [1, k]$ and k and k is the number of answers extracted for each context $\hat{c}_j$. The generated answers, along with the contexts, $(\hat{a}_i, \hat{c}_j)$, are then given as input into the QA model, which generates corresponding questions $\hat{q}_m$, where $m \in [1, k]$. This technique is repeated for each context in the document, yielding a collection of created *context-question-answer* triples. Finally, utilising the synthetically created triples as domain-specific training data, the pre-trained SQuAD-based EQA model is fine-tuned. It is vital to note that *QASAR* is an automated self-learning system that automatically generates training data using language models, avoiding the requirement for costly annotated training data.

The pipeline for the online training phase can be summarised as outlined in Algorithm 6.1.

6.2.4 *QASAR* Inference Time

Our proposed *QASAR* framework's final *inference step* is essentially a real-time user interaction interface. The QA platform is fully trained and ready to answer user queries after the user provides the required domain-specific documents, and the online fine-tuning process is completed. At inference time, as depicted in Figure 6.2, a question q_{IT} is received from the user. The SBERT-

Algorithm 6.1: Pipeline for the online fine-tuning of *QASAR* for domain adaptation.

1. **Passage Retrieval**: Extract a paragraph/context based on *document chunking*.
2. **Answer Extraction**: Extract answers from the obtained contexts using the T5 pre-trained Answer Extraction Model.
3. **Question Generation**: Generate questions, from answers extracted and the associated context, using the T5 pre-trained duestion generation model.
4. **Fine-Tuning**: Fine-tune a pre-trained SQuAD-based QA model on the generated synthetic context-question-answer triples.

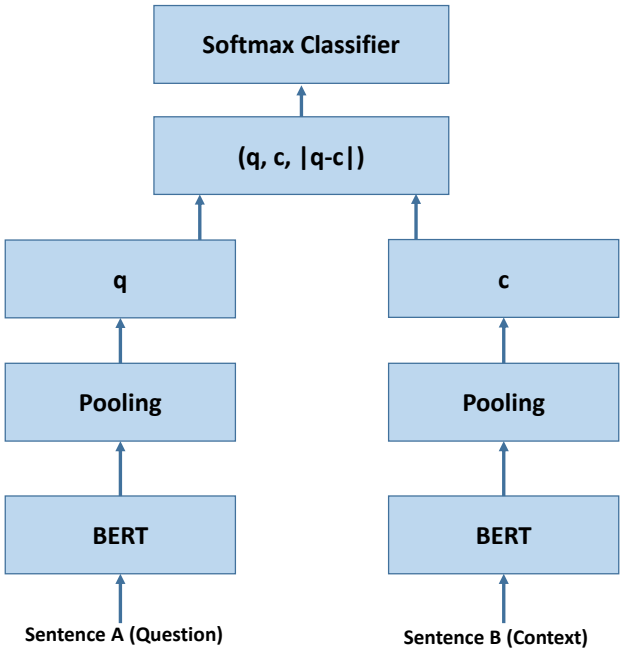

Figure 6.3: SBERT Training Architecture with classification objective function (Adapted to Question Answering) [103].

based Context Retrieval model, trained during the offline phase, uses the semantic meaning of the query Q_{IT} to extract a candidate context c_{IT} that contains the potential answer along with supporting information. This enables the EQA system to extract the answer span from the provided context. The final step involves providing the context-question pair, (q_{IT}, c_{IT}), to the fine-tuned EQA model that was trained on the synthetically generated domain-specific dataset for final answer extraction.

Suppose the user submits further documents related to a different domain. In that case, the same *online fine-tuning* step is re-run on the models from fine-tuned during the offline training phase, automatically adapting the EQA model to the new domain via self-learning.

6.2.5 *QASAR* Model Training Setup

In this section, we introduce the overall setup of the 4 models of *QASAR*, and the details of the training performed in the offline phase.

6.2.5.1 Answer Generation:

The purpose of the Answer Generation model in the *QASAR* framework is to generate potential answers, given a context ($Pr(a|c)$), that reflects the prior distribution of answers from other datasets. One approach is to consider named entities and noun phrases as answer candidates, but this has been shown to perform worse compared to an explicitly learned answer extraction module [98]. To achieve this, we fine-tune a T5 model using the text-to-text format. However, unlike fine-tuning a BERT architecture for question-answering, following Alberti et al. [5], we do not include the question tokens in our training as the goal is to extract likely answers only from the context. Although BERT-based answer extraction strategies [98] predict the start and end positions of the answer token span $(s, e) = a$, we formulate it as a text generation objective utilising the T5 model. We utilise the T5 model for this task due to its large number of parameters and strong empirical performance on text generation tasks [101].

We use the following procedure (inspired from [17]) for pre-processing the SQuAD training dataset to be provided as input to the T5 model training:
(a) Each context in the training data is initially divided into individual sentences (using standard sentence splitting tools like spaCy [43]).
(b) Sentences containing ground-truth answer(s) are then flagged by highlighting them with the special token <HA>.
(c) Finally, for the target text and in case there is more than one answer, we separate them using <sep> token.

For example, consider the following training sample from the SQuAD dataset, containing a context and related answers (of questions on the context):

> *Context:* On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. The $1.2 billion stadium opened in 2014. It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003.
> *Answers:* May 21, 2013; 2014; $1.2 billion; San Diego; and Boston.

Hence, for this instance, since all the sentences contain one (or more) answers, they are padded with the special token and the answers are concatenated together with the <sep> token. The processed training data is obtained as:

> *Input text:* <HA> On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. <HA>The $1.2 billion stadium opened in 2014. <HA> It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003.
> *Output text:* May 21, 2013 <sep> 2014 <sep> $1.2 billion <sep> San Diego <sep> Boston.

After processing the SQuAD training data in the above format, we tune a T5 model to obtain the *Answer Generator* module of QASAR. Specifically, in our setup, we tune the *t5-small* architecture[7] with a batch size of 32, learning rate set to $1e^{-4}$, and for 10 epochs. The parameters were obtained using grid-search.

6.2.5.2 Question Generation:

The question generation component in *QASAR* is trained as a conditional language generation model, $Pr(q|a, c)$, using a T5 model. The training process involves highlighting the answer span

7 Available at https://huggingface.co/t5-small.

within the context using the special token <AN>, where the context and answer pair, (c, a), are treated as input. This enables the model to understand the context to be utilised for question generation. The training data is derived from pre-processing the SQuAD dataset per the following methodology:

For each training context in the SQuAD dataset, there exist multiple questions (5 in the above example) with their corresponding answer. Hence, in our setup, we create five different samples, each having the same input text but the output text would represent the different questions. Finally, all the training samples, thus obtained, are concatenated and the *Question Generator* model is trained using a $T5-small$ architecture and the same parameter setting as that of the above *Answer Generation* model training.

Input text: On <AN> May 21, 2013 <AN>, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's Stadium. The $1.2 billion stadium opened in 2014 . It is the first Super Bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003.
Output text: When was Levi's Stadium awarded the right to host Super Bowl 50?

6.2.5.3 Extractive Question Answering:

In this module, we consider a typical EQA model pre-trained on SQuADv2 for predicting the correct answer span for a question given a relevant passage or context. There already exist various pre-trained language models fine-tuned towards open-domain QA based on SQuAD training dataset. Hence, in *QASAR*, we do not train an EQA model from scratch but directly employ the pre-trained *SpanBERT-SQuAD*[8][45] model (based on span-oriented pre-training objective for answer span representation). Alternate EQA models, such as RoBERTa-SQuAD[9][72] (built upon fine-tuned BERT architecture with a larger vocabulary) can also be easily used in our framework.

6.2.5.4 SBERT Context Retrieval:

Contextualised language models, such as BERT, have been found to generate dense word embeddings that effectively capture the semantic meaning of text, as contextual information is incorporated in the model [22]. However, a limitation of the BERT network architecture is the lack of explicit computation of independent sentence embeddings. To address this, methods like averaging the output word vectors or using the "[CLS]" token representation have been proposed for deriving a fixed-size sentence representation for passage retrieval [162, 79]. However, these methods have been shown to generate relatively poor sentence embeddings [103], which are often inferior to the simple averaging of context-free embeddings like GloVe [97].

In an attempt to bridge such gaps, *Sentence-BERT* (SBERT) was introduced to produce enhanced fixed-length sentence embeddings – demonstrating state-of-the-art results on various unsupervised learning tasks [103]. SBERT, a modification of the pre-trained BERT architecture, employs a Siamese network to generate semantically meaningful sentence embeddings that can be compared using the cosine similarity measure. In our framework, we utilise the SBERT architecture for our *context retrieval* module, to retrieve relevant candidate contexts from a large document (or article) that is semantically closer to the user query and might possibly contain the answer.

8 https://huggingface.co/mrm8488/spanbert-finetuned-squadv2
9 https://huggingface.co/deepset/roberta-base-squad2

QASAR uses the SBERT-BERT-NLI model[10] which is pre-trained on the NLI [15] corpus. NLU corpus is a collection of sentence pairs considered as *premise-hypothesis* and annotated with labels *contradiction*, *neutral* or *entailment*. To adapt the SBERT model for QA context retrieval, we further fine-tune it on the SQuAD dataset by adopting on the bowman15nli training format, as follows. For each question in the SQuADv2 training dataset, we extract sentences from the associated context to construct question-context sentence pairs (akin to the premise hypothesis setting). If the context sentence contains the ground-truth answer, the question-context sentence pair is labelled as *entailment*, otherwise is considered to be *contradiction*. Hence, the question-sentence pair (q, c_s) is considered as entailment (positive instance) if the answer $a_i \in c_s$ and c_s denotes a sentence of the context, else considered as a contradiction (negative instance). This setup enables the model to understand the difference between the positive and negative examples by contrasting the semantic information present in them.

In this setting, SBERT embeddings of questions and context sentences along with the above obtained question-context sentence-pair labels are used to fine-tune the SBERT model (for QA setting) based on the classification objective function, as shown in Figure 6.3. Mathematically, sentence embeddings of question q and context-sentence c_s are concatenated with the element-wise difference $|q - c_s|$ to train a *softmax* classifier for learning the weight matrix $W_t \in \mathbb{R}^{3n \times k}$, as

$$o = \mathrm{softmax}(W_t(q, c_s, |q - c_s|)) \tag{6.1}$$

where n is the dimension of the SBERT sentence embeddings and k is the number of labels.

6.3 EXPERIMENTAL SETUP

In this section, we introduce the empirical setup that we used to evaluate our proposed *QASAR* framework and compare its performance to other existing baselines on QA datasets.

6.3.1 Datasets

Datasets in the field of QA are often categorised into two types: *open-domain* and *closed-domain*. Answering questions concerning generic text and general knowledge using a standard language, such as Wikipedia articles and news reports, is referred to as open-domain QA. These scenarios can also make use of external sources of common knowledge, such as ontologies, and have a greater amount of data accessible for training QA systems.

Closed-domain question answering, on the other hand, involves queries related to a particular field with a separate vocabulary, such as medical, automobile mechanics, or law. This sort of QA is more difficult since training resources are limited, and QA models trained on open-domain data do not generalise effectively.

Our proposed framework, *QASAR*, employs *self-supervised learning*, which does not require domain-specific labelled training data and can be trained offline using only SQuAD data, as described in Section 6.2. We evaluate the generalisation capability of *QASAR* on open-domain datasets, similar to SQuAD, as well as on closed-domain QA datasets from the fields of biomedical and science domains. To this end, we use the following datasets in our evaluation:

10 huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

Table 6.1: Test Dataset Characteristics

Dataset	# Contexts or Docs	# Questions	Avg. # Sent. per Doc	Domain
NQ	1742	1772	4.31	Open
DROP	281	1260	9.42	Open
NewsQA	638	4185	30.64	Open
BioASQ	451	142	10.10	Biomedical
TextbookQA	389	1499	41.21	Science

6.3.1.1 Open-domain QA datasets:

- *Natural Question* (NQ) [55] – The NQ dataset was developed by Google and is comprised of real, anonymised user queries submitted to the Google search engine. To create the dataset, an annotator was given a question and the top 5 Wikipedia pages from the search results. If the answer to the question could be found on the page, the annotator marked a long answer and a short answer. If no answer was found, it was labelled as "null". The authors of the NQ dataset claim that the "unanswerable" questions in SQuAD can be easily identified, while the NQ task requires more complex reasoning to determine whether a paragraph contains the answer.

- *NewsQA Dataset* [127]: Microsoft has created the NewsQA dataset, which comprises 100,000 human-generated question-answer pairs generated from over 10,000 news articles from CNN. The dataset was created through crowd-sourcing, where workers generated questions and answers based on the articles. The NewsQA dataset presents several challenges for question-answering models, including answers of arbitrary length, a lack of answers for some questions, the absence of candidate answers for annotators, and a significant number of questions that require reasoning beyond simple word and context matching. Although some of these challenges are also present in SQuAD, the authors of [127] have shown that NewsQA presents a greater challenge for EQA models than SQuAD.

- *DROP Dataset* [25]: The Discrete Reasoning over Paragraphs (DROP) dataset is introduced in an effort to focus more on questions that require some textual reasoning to reach the correct answer. The questions in this dataset are inspired by the sophisticated, compositional questions seen in the semantic parsing literature (on Wikipedia articles).

6.3.1.2 Closed-domain QA datasets:

- BioASQ [87] conducts challenges focused on semantic indexing and QA in the biomedical domain. Participating systems are tasked with providing answers as part of the evaluation process, which are subsequently examined by domain experts. In our experimentation, we utilise 30% of the recent Bio-Medical Semantic QA dataset for COVID-19 from the BioASQ competition.

- The TextbookQA (TQA) dataset [49] consists of questions derived from a middle school scientific curriculum encompassing life science, earth science, and physical science. The

authors observed that conventional EQA approaches exhibit lower performance on the TQA dataset compared to SQuAD, indicating that TQA presents unique challenges due to its high domain-specificity.

Table 6.1 describes the characteristics of the individual datasets. For all the datasets[11], we have not considered any training instance, as the major aim of this contribution is to eliminate the need for annotated data – thus making QA systems generalizable to both open- and closed domains. More specifically, we do not utilise samples from the training split, rather we utilise the contexts from the test set to generate synthetic QA pairs for fine-tuning the models.

6.3.2 Baseline Methodologies

We benchmark the performance of our proposed framework against the state-of-the-art open-domain QA model *SpanBERT-SQuAD*, pre-trained on SQuAD dataset. The *self-learning* based fine-tuned (on synthetically generated data) variant of this model (denoted as *SpanBERT-SQuAD-GQA*) is also considered as a baseline to access the impact of *self-learning* on domain adaptability.

With regards to the context retrieval module (for extracting relevant passages for EQA), we compare the following methods:

- *Google DialogFlow* (DF) – this system is a widely used industrial solution for building enterprise-level chatbots. It works by receiving a document (known as a knowledge base) and a user's inquiry, then delivering a set of relevant sentences (i.e., context) to help with the task of answering inquiries.
- *BM25* [34] – refers to the traditional information retrieval system based on term overlap and frequency (between question and context in this setting);
- *Re-ranker* [36] – ranking framework fine-tuned from language model by estimating candidate relevance based on rich contextualised signals[12]; and,
- *SBERT* – a sentence embedding approach encapsulating both semantic and contextual information of texts. The similarity between the texts can then be determined by computing the cosine similarity between their respective embeddings, making it suitable for utilisation in retrieval tasks[13]. This is different from the SBERT we use in QASAR, as we use SBERT fine-tuned on the synthetic dataset in our proposed approach.

For a fair comparison, we set the baselines (BM25, Re-Ranker and SBERT) to retrieve the same number of sentences as context for EQA. DF is an off-the-shelf platform without any such user parameter choice, however we observed it to return contexts similar in size to our experimental setup, except for NewsQA and TextBookQA containing large contexts.

6.3.3 Evaluation Metrics

To compare the performance of the different QA frameworks on open- and closed-domain datasets, we evaluate on the following two widely used measures in the literature:

11 All datasets have been obtained from github.com/mrqa/MRQA-Shared-Task-2019#mrqa-2019-shared-task-on-generalization, except NQ which is taken from github.com/google/retrieval-qa-eval/blob/master/nq_to_squad.py.

12 github.com/luyug/Reranker

13 www.sbert.net/examples/applications/information-retrieval/README.html

- *Exact Match* (EM) – Given a predicted answer $\hat{a}$ by a QA systems and the ground-truth answer a (for a question q), the EM measure computes if the predicted answer exactly matches the original answer. The number of such exact matches is computed for the entire test suite, and the average is reported as the EM score. This essentially provides the *accuracy* for EQA scenarios.

- *Token-level F1 Score* (F1) – This measure the precision-recall based F1 score using the number of overlapping tokens between $\hat{a}$ and a, i.e., the number of tokens common between them. In this measure, individual words present in the answer span are considered as tokens. The average F1 score over all questions is reported as the performance.

All experiments and model fine-tuning were performed using 2 GeForce GTX 1080Ti GPUs, each with 8GB memory and CUDA v10.2. The context retrieval module was set to extract around 50% of the input document as relevant context for the question.

6.3.4 Implementation Details

In our experiments, we optimise the *cross-entropy* based loss, with $n = 768$ and $k = 2$ (positive entailment and negative contradiction examples from SQuAD). We randomly sample at most 3 negative instances (context sentences that do not contain the answer) for constructing a well-balanced dataset. This is used to fine-tune the SBERT architecture with a softmax classifier (see Figure 6.3) for 5 epochs, using a batch-size of 8, Adam optimiser with learning rate $1e^{-6}$, and a linear learning rate warm-up over 10% of the training data with the mean pooling strategy. The hyperparameters values were selected by evaluating the performance over the validation set.

6.4 RESULTS AND DISCUSSION

This section presents the empirical results obtained by *QASAR* and other baseline QA systems on the different datasets.

6.4.1 Domain Adaptation.

The central concept of the *QASAR* framework is *self-supervised domain adaptation*, which leverages synthetically generated domain-specific data. In order to evaluate this approach, we examine the performance enhancement achieved through fine-tuning pre-trained QA systems based on language models using the generated training QA pairs on domain documents in the *online fine-tuning phase*, as previously described in Section 6.2.3.

The results in Table 6.2 demonstrate the strength of the fine-tuned *SpanBERT-SQuAD-GQA* model obtained via the *QASAR* framework's self-supervised domain adaptation method, compared to the pre-trained *SpanBERT-SQuAD* model. It was found that the self-supervised domain learning leads to significant improvement in both open- and closed-domain QA settings, with an increase of up to **10.0** points in both EM and F1 metrics. These results validate the efficacy of the proposed self-learning approach for domain adaptation and indicate that the quality of the synthetic dataset generated is high and relevant. As a demonstration, the *answer and question*

Table 6.2: Performance of Fine-tuned Domain-adapted SpanBERT-SQuAD EQA model of QASAR.

Model	NQ		DROP		NewsQA		BioASQ		TextBookQA	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SpanBERT-SQuAD	35.21	49.71	24.08	29.91	37.93	53.19	38.27	49.04	28.94	37.00
SpanBERT-SQuAD-GQA‡	**45.18**	**60.95**	31.71	40.99	**40.42**	**57.40**	46.94	58.88	33.46	44.81
QASAR‡	44.18	59.76	**32.67**	**41.06**	40.21	56.63	**52.21**	**63.70**	**35.32**	**47.02**

‡ Results are *statistically significant* with $p < 1e-5$ based on paired t-test with *SpanBERT-SQuAD*.

generation modules using the T5 model were applied to the BioASQ bio-medical domain dataset, yielding examples of synthetic domain-specific training data.

Context: The patient was interpreted as the primary active acute EBV infection. A skin biopsy showed leucocytoclastic vasculitis. The other viral and bacterial investigations were negative.
Questions: What did a skin biopsy show?; What were the other viral and bacterial investigations?
Answers: leucocytoclastic vasculitis; negative

We analyse the results presented in Table 6.2 and observe that the integration of the *QASAR* framework, including the fine-tuned EQA model and context retrieval module, leads to improved performance compared to only utilising the fine-tuned EQA model in closed-domain datasets. On open-domain datasets, the results are comparable. This improvement on the closed-domain datasets can be attributed to the contextual refinement provided by the SBERT context retrieval module, which effectively reduces noise and irrelevant information that might negatively impact the performance of the EQA model.

6.4.2 Context Retrieval Performance Comparison.

The performance of the SBERT-based context retrieval module in the *QASAR* framework was evaluated by comparing it to other methods using the entire EQA pipeline. The results, presented in Table 6.3, demonstrate that the EQA system that integrates the SBERT context retriever, fine-tuned on the SQuAD training data through the NLI classification framework, outperforms all baselines. This exemplifies the effectiveness of the fine-tuned context retrieval model in comprehending the semantic similarity between questions and contexts, resulting in the retrieval of more relevant and less noisy candidate contexts. The provision of improved context information to the EQA module enhances the overall performance of the *QASAR* EQA framework.

A comparison was made between the SBERT-based Context Retrieval module in the *QASAR* framework and other methods by evaluating the entire QA pipeline (context retrieval + EQA). The results, presented in Table 6.3, demonstrate that the QA system with the proposed SBERT context retriever, fine-tuned on SQuAD training data using the NLI classification framework, outperforms all the baselines. This highlights the advantage of the fine-tuned context retrieval model in understanding the semantic similarity between questions and contexts, leading to the retrieval of more relevant candidate contexts with reduced noise and irrelevant information. The improved context information supplied to the EQA module results in improved performance of the overall *QASAR* QA framework.

Table 6.3: Comparison of different Context Retrieval Methods for EQA on Performance across the datasets.

Model	Context Retriever	NQ		DROP		NewsQA		BioASQ		TextBookQA	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SpanBERT-SQuAD	DF	32.73	45.88	22.02	27.84	32.50	45.41	35.39	51.20	29.14	36.74
	BM25	31.65	45.07	23.68	29.81	35.68	50.03	37.16	46.98	27.01	35.28
	ReRanker	34.82	49.08	25.14	31.29	36.79	51.52	40.93	52.03	28.67	37.31
	SBERT	33.06	46.77	24.28	30.48	34.54	48.46	38.92	48.24	27.87	36.38
	SBERT-QA	**35.27**	**49.17**	**25.54**	**31.42**	**37.58**	**52.64**	**42.47**	**53.91**	**29.40**	**38.01**
SpanBERT-SQuAD-GQA	DF	39.72	53.55	24.48	31.39	32.85	46.18	50.44	63.09	34.73	43.80
	BM25	41.19	56.07	30.87	39.85	38.48	54.33	46.46	56.47	33.00	44.43
	Re-Ranker	**44.30**	**60.03**	31.53	40.28	39.31	55.34	50.00	62.49	34.73	46.26
	SBERT	42.21	57.43	31.86	40.65	37.65	53.25	44.47	54.93	33.99	45.98
	SBERT-QA*	44.18	59.76	**32.67**	**41.06**	**40.21**	**56.63**	**52.21**	**63.70**	**35.32**	**47.02**

* Observe *SpanBERT-SQuAD-GQA* + *SBERT-QA* comprises the overall *QASAR* framework.

The *Re-ranker*, serving as the context selector, demonstrated strong overall performance, including for closed-domain datasets, according to the results. Conversely, other methods, including the industrial Google DialogFlow platform, performed poorly across the datasets. This poor performance was attributed to a lack of semantic and contextual relevance between the collected context and the user's question, leading to a decrease in the overall performance of the QA system.

Interestingly, in certain situations, the use of the context retrieval module, might result in a slight decrease in the overall performance (e.g., *NQ* dataset in Table 6.2). Such a marginal drop in performances can be attributed to two reasons:

- The original context size is itself very small, and hence the further selection of a smaller sub-context (for the EQA module) might prune out the necessary information. For instance, on average a document in NQ contains only ~ 4 sentences (Table 6.1) – thus any further context reduction might negatively impact the performance; and,

- Complex multi-hop reasoning is necessary for answering a question, and hence even seemingly irrelevant information is required in such a dataset. Semantic similarity (between question and context) based approaches would fail to identify such long-distant dependencies, and prune them as noise – leading to a performance dip.

In such cases, the amount of context retrieved must be properly tuned depending on the QA dataset characteristics (as further discussed in Sec. 6.4.3.2).

6.4.3 Qualitative Study

In this section, we analyse the robustness of our *QASAR* QA framework on diverse experimental parameter settings on different datasets. Specifically, we study the following scenarios:

6.4.3.1 Passage Length for Self-Learning:

Synthetic domain-specific training data created by the *Answer Generation* and *question generation* modules during *Online Fine-Tuning* phase of QASAR, depends on the number of sentences (l) in the individual contexts $\hat{c}$ that the input document(s) is divided into (refer Sec. 6.2.3). This

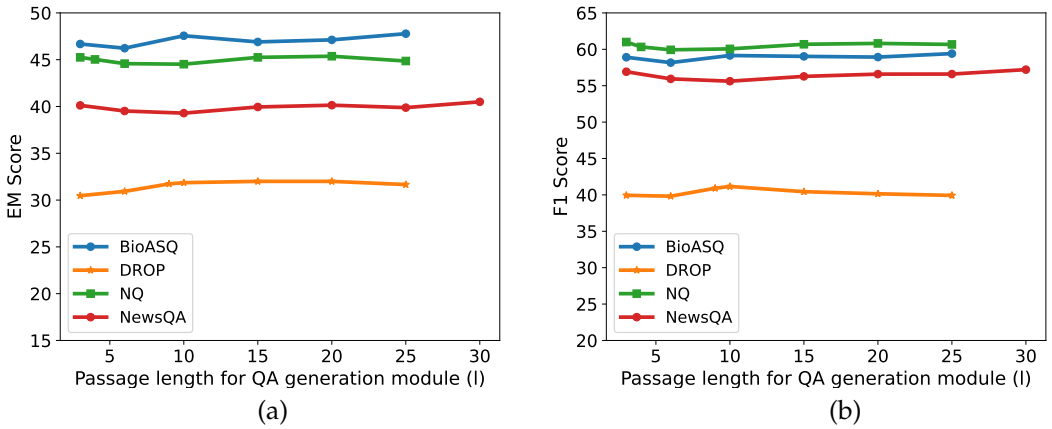

Figure 6.4: Performance impact of *QASAR* with varying Passage Length (l) for Self-Learning: (a) EM and (b) F1 scores.

may impact the fine-tuning of the EQA model on closed-domain QA datasets, and subsequently, affect the overall performance of QASAR.

To this end, we vary the value of l (i.e., the number of sentences forming the context for domain adaptation in the online phase) and report the observed performance in Figure 6.4. It can be observed that the proposed self-learning module is fairly stable against such variations, with *QASAR* depicting similar overall performance across the datasets. In our experiments, we set the parameter l to the average length of the domain-specific document(s) provided by the user (i.e., the average number of sentences per document). For example, in BioASQ we set, $l = 10$ using the value from Table 6.1.

6.4.3.2 SBERT Context Retrieval Length:

Another important module of *QASAR* is the SBERT-based context retrieval module that is used during the *inference* phase – extracting relevant contexts pertaining to a user question (q_{IT}) to be provided to the fine-tuned EQA model for extracting the final answer token span (from the provided context). Setting the number of sentences (k) retrieved by the SBERT-context module is crucial, as larger context sizes would induce noise and irrelevant information, while smaller context lengths would eliminate vital information for finding the answer. Both scenarios would degrade the performance of the EQA module of *QASAR*, and hence we empirically set this hyper-parameter for the different datasets.

As seen from Figure 6.5, the performance of *QASAR* approximately demonstrates a bell-shape (following the above intuition) with varying sizes of the context extracted by the SBERT context retrieval module. Hence, the context length needs to be carefully set for the different datasets, depending on the value which provides the best results.

6.4.3.3 Impact of EQA model:

We now study the generalisability of our *self-learning* approach on different existing EQA frameworks. To this end, we replace the SpanBERT-SQuAD EQA model of *QASAR* with RoBERTa-SQuAD, and appropriately fine-tune it on the synthetically generated data for domain adaptation (as discussed previously in Sec. 6.2.3). From Table 6.4, we observe that the *self-learning* based

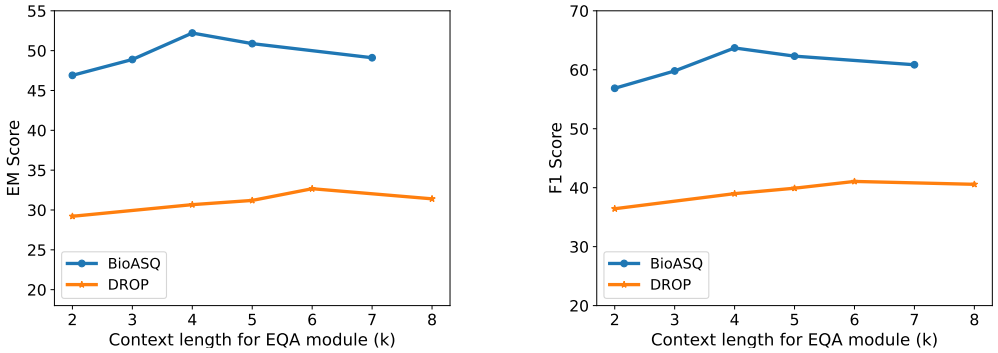

Figure 6.5: Performance impact of *QASAR* with varying context size (k) for SBERT Context Retriever: (a) EM and (b) F1 scores.

Table 6.4: Domain Adaptation results on SpanBERT-SQuAD and RoBERTa-SQuAD EQA models.

Model	NQ		BioASQ	
	EM	F1	EM	F1
RoBERTa-SQuAD	46.22	58.50	37.38	46.58
RoBERTa-SQuAD-GQA	**46.33**	**61.43**	**46.46**	**59.34**
SpanBERT-SQuAD	35.21	49.71	38.27	49.04
SpanBERT-SQuAD-GQA	**45.18**	**60.95**	**46.94**	**58.88**

fine-tuning improves the performance for both the EQA models – depicting that our proposed framework is robust and effective across different model choices.

6.4.3.4 Few-shot Learning:

The cost of acquiring a sizable, domain-specific labelled training dataset can be prohibitive when assessing a QA system for a new topic. As an alternative, few-shot learning can be used to manually produce a limited number of annotated training instances and utilise them for domain adaptation of the EQA module. The purpose of this study is to look into how using few-shot learning impacts the performance of *QASAR* when adapting to a new domain. The evaluation findings showcase the stability of the performance of *QASAR* across datasets, as shown in Figure 6.6. This consistency is a sign of the superior quality of the synthetic dataset produced using the self-learning methodology, as detailed in Section 7.2.1. The synthetic dataset's superior quality is further supported by the close approximation of the answer span distribution of ground-truth annotations, especially in closed-domain datasets as there is only a slight change in model performance on tuning on the few-shot instances.

We compare the performance of SpanBERT-SQuAD fine-tuned on both the synthetic data generated and the limited annotated training data to further emphasise the advantages of the suggested *self-learning* approach. The performance of SpanBERT-SQuAD fine-tuned solely on a small number of annotated training examples (referred to as FS) is compared against SpanBERT-SQuAD fine-tuned on both the synthetic and limited annotated training data. The comparison's

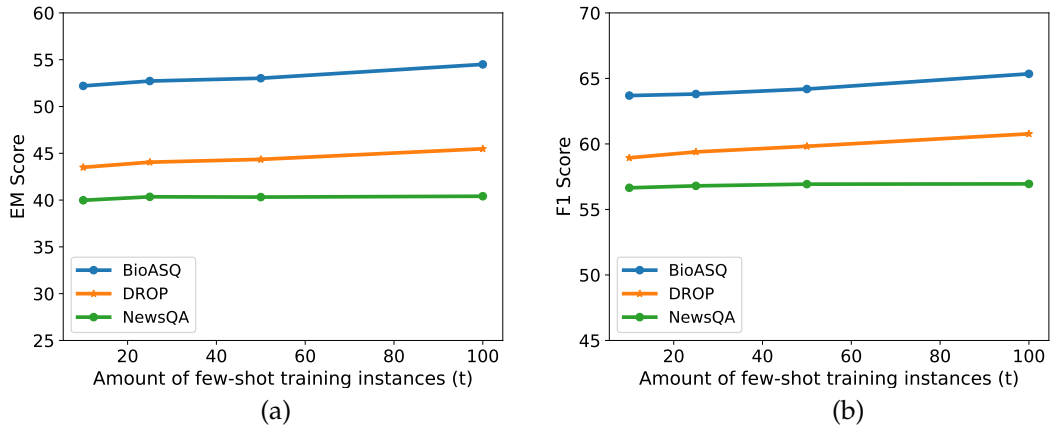

Figure 6.6: Performance impact of *QASAR* with few-shot training (t) instances: (a) EM and (b) F1 scores. In this setting, the EQA component of *QASAR* is fine-tuned on the few-shot training instances.

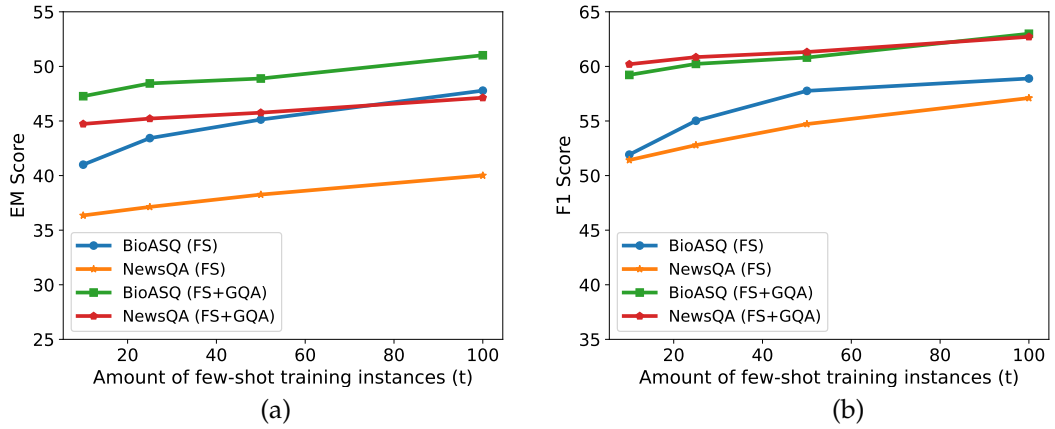

Figure 6.7: Performance impact of *self-learning* and *few-shot*: (a) EM and (b) F1 scores. In this setting, the online fine-tuning of the EQA component involves tuning on the combined synthetic and few-shot training instances

findings are presented in Figure 6.7 which demonstrates how the self-learning framework significantly enhances the domain adaption of generic QA systems even in situations with a limited number of annotated training examples. We posit the self-learning approach to be the *default standard* for QA models due to its low cost, regardless of whether the models are meant for open- or closed-domain applications.

6.4.3.5 *Multi-task Learning:*

We study a variation of the *self-learning* procedure as employed in QASAR. Note that, we have two independently pre-trained T5 language models – one for answer generation, and the other for question generation. However, in certain enterprise production environments, this might cause issues due to its large memory requirements and training time. To this end, a variation is to use a single T5 model that is trained for multi-tasking, i.e., a single model is trained for both answer generation and question generation. We explore this scenario by using a single

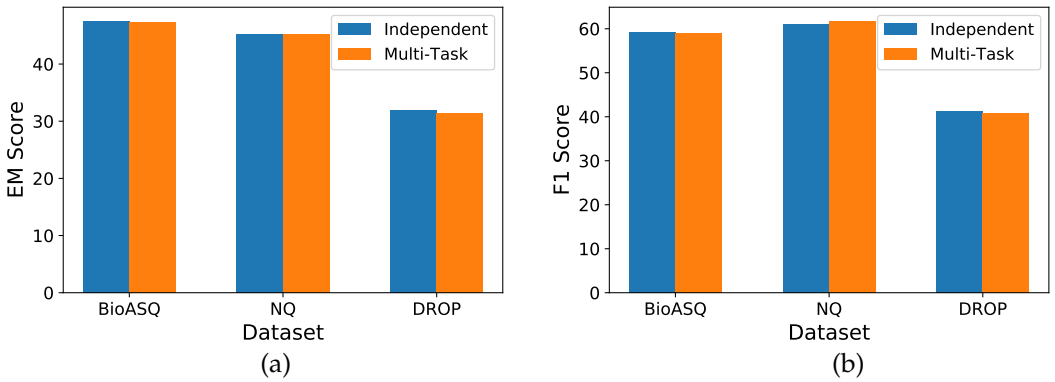

Figure 6.8: Performance impact of *QASAR* with T5 multi-task learning: (a) EM and (b) F1 scores.

multi-tasking model in QASAR. As observed from Figure 6.8, *QASAR* exhibits robustness to such settings as well, with no noticeable impact on the overall performance.

In summary, from the above results, we observe that the proposed *QASAR* QA framework for *self-learning* based domain adaptation is extremely robust to varying empirical settings, amenable to different model choices, and efficient across various open- and closed-domain datasets – significantly outperforming existing QA models.

6.5 LINKING EQA AND AS2

It is crucial to acknowledge that not all user queries can be resolved by selecting spans of text from unstructured sources. The AS2 component is designed to handle situations where answering a user's query requires an entire sentence or multiple sentences. Chapter 7 is dedicated to describing the self-supervised training of the AS2 module.

7 KNOWLEDGE SELECTION FROM UNSTRUCTURED SOURCES IN CHATBOTS: ANSWER SENTENCE SELECTION

Selecting an appropriate and relevant context forms an essential component for the efficacy of several information retrieval applications like *QA* systems. The problem of *AS2* refers to the task of selecting sentences, from a larger text, that are relevant and contain the answer to users' queries. While there has been a lot of success in building AS2 systems trained on open-domain data (e.g., SQuAD, NQ), they do not generalise well in *closed-domain* settings, since domain adaptation can be challenging due to poor availability and annotation expense of domain-specific data. This section proposes *Self-lEarning for Domain adaptation in Answer seNtence selection* (SEDAN), an effective self-learning framework to adapt AS2 models for domain-specific applications. We leverage large pre-trained language models to automatically generate *domain-specific QA pairs* for domain adaptation. We further fine-tune a pre-trained Sentence-BERT architecture to capture semantic relatedness between questions and answer sentences for AS2. Extensive experiments demonstrate the effectiveness of our proposed approach (over existing state-of-the-art AS2 baselines) on different QA benchmark datasets.

7.1 INTRODUCTION

AS2 involves the selection of sentences that contain the information required to answer a given question and forms an important problem in its own right. It enables several applications like information search, snippet generation, summarisation and knowledge base generation, as well as in the larger context of open domain question answering [157]. The relevance of an answer sentence to a question is typically determined by measuring the semantic similarity between the question and answer. Traditional systems like BM25 [6] were initially used to gauge information relatedness between questions and answers based on the bag-of-words (BoW) model. With the advent of neural models [20, 57, 37, 58, 39], such architectures have been shown to perform extremely well in the AS2 task. Transformer [130] based language models such as BERT [22] and RoBERTa [72] have enabled the capture of the contextual relationship between words within a sentence, to achieve state-of-the-art results in various natural language processing tasks. Such LMs have also been successfully adapted to the AS2 task [37, 57, 58].

Despite the growing interest in QA systems, the use of large documents such as Wikipedia articles in these systems presents a challenge due to the presence of diverse contextual information that may be irrelevant to the specific question being asked. The presence of such a large "noisy" context might adversely affect the performance of the QA pipeline. Hence, identifying potentially relevant text snippets from a large context becomes pivotal for enhanced QA

Q	**Which drug should be used as an antidote in benzodiazepine overdose?**
S1	*A 54-y-old man ingested 2 g of bulk laboratory diazepam and was treated with activated charcoal, enhanced diuresis and flumazenil infusion.*
S2	The treatment resulted in awakening, but the patient had drowsiness, dysarthria, diplopia, and dizziness for 9 d. Blood levels of diazepam and its main metabolite, nordiazepam, were obtained for 1 mo.
S3	The half-lives in this benzodiazepine overdose were longer than those seen with therapeutic doses.
S4	Benzodiazepines should not be readministrated when patients awake after suicide attempts.

Table 7.1: Challenges in domain-specific AS2. Only sentence S1 is relevant to query Q and requires understanding of relation between *D*iazepam & *B*enzodiazepine.

performance – motivating the need for efficient and accurate AS2 methodologies. In fact, the reduction of "noisy" irrelevant context (w.r.t. to a question) has been shown to improve QA performance (Section 6.4.3.2) in production settings [37].

This challenge is more compounded in *domain-specific* settings due to the need for specialised understanding in terms of domain semantics, terminology, and relationships. For example, Table 7.1 shows a QA scenario from the biomedical domain. Given question **Q**, we can observe that only sentence **S1** is pertinent for answering the question. Sentences S2, S3 and S4, although relevant within the domain, are incapable of answering **Q**. Note, that an AS2 system in this scenario needs to understand the implicit relation that `Diazepam is a type of Benzodiazepine` to identify the answer sentences correctly. This depicts the need for *domain-aware AS2 models* for subsequent improvements of end-to-end QA platforms.

Domain adaptation has been relatively unexplored in the context of Answer Sentence Selection (AS2). The recent work of Garg et al. [37] suggested a two-step fine-tuning of BERT-based architecture for domain adaptation. However, such models require expert-annotated domain-specific data, wherein the creation of such datasets can be a time-consuming and expensive process. Additionally, the availability of domain experts can be challenging in several specialised areas and enterprise settings. Furthermore, to the best of our knowledge, the impact of using AS2 models on different downstream tasks such as Extractive Question Answering (EQA) has not yet been explored.

In this section, we propose the SEDAN framework to ameliorate the above issues. We show that *SEDAN* can automatically adapt itself to a specific domain via *self-learning* without the need for any explicitly annotated training data. We utilise fine-tuned *Sentence-BERT* architecture to capture the semantic relationships between a question and answer texts – for *SEDAN* to extract appropriate answer sentences in closed-domain as well as open-domain settings. We further study the performance impact of *SEDAN* (and other baselines) on the downstream EQA task. In summary, our contributions are:

- *SEDAN*, a *self-learning* based framework for Answer Sentence Selection (AS2) is proposed that achieves domain adaptation by using synthetically generated QA pairs from domain-specific documents by leveraging large pre-trained language models.
- A Siamese network-based fine-tuning of Sentence-Transformers (S-BERT) is proposed for capturing the semantic understanding between questions and answer sentences.

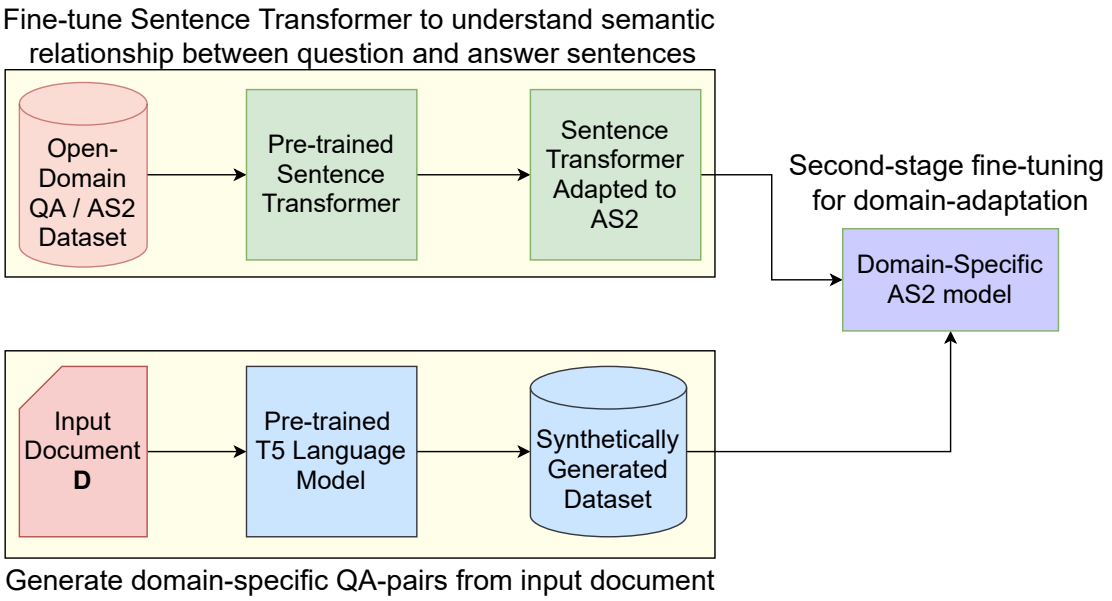

Figure 7.1: Modular overview of proposed framework, *SEDAN*. The first step involves fine-tuning transformer model on a generic QA dataset (e.g., SQuAD). The second fine-tuning utilises domain-specific synthetic QA pairs generated using pre-trained large language model (e.g., T5).

- Extensive evaluation is conducted to demonstrate strong *evidential empirical performance improvements* of *SEDAN* (over other baselines) on AS2 task, as well as in the downstream task of EQA on multiple benchmark datasets.

7.2 METHODOLOGY

In this section, we now formally introduce the problem and describe the proposed *SEDAN* framework. We discuss how domain-specific QA pairs are automatically generated for domain adaptation and the subsequent fine-tuning of Sentence-BERT for suiting the AS2 task.

Task Definition: Given a question, AS2 involves selecting the answer sentence(s) from a large context passage. Answers are those sentences in the context passage that potentially contain the answer to the question posed. More formally, given a context passage C_i consisting of sentences $S_i = \{S_{i_1}, S_{i_2}, \cdots S_{i_n}\}$ and a question Q pertaining to the context passage C_i, the aim is to learn a ranking function $f : Q \times S_i \rightarrow \mathbb{R}$ that assigns a score to each of the Question-Answer sentence pairs, where a higher score indicates a higher probability of the sentence containing the answer of Q.

7.2.1 Synthetic Dataset Generation

The synthetic dataset generation module is an essential component of *SEDAN* for *self-learning*. The synthetic data generation in *SEDAN* is similar to that detailed in Section 6.2.2 (can also be replaced with a different data generation scheme). We use a pre-trained T5 model to generate QA pairs from a domain-specific document D. Since transformer models can process only 512 tokens at a time, we split the document into chunks of K_{GQA} sentences, to prevent the T5 model

from ignoring document text, while preserving domain-specific contextual information. Once the data is chunked into multiple short contexts, we follow a two-step process to generate QA pairs.

Answer Generation Given the document, D, chunked into contexts, $C_1, C_2, \cdots, C_n$, we use each C_i to generate QA pairs. Similar to Puri et al. [98], given a context passage C_i, we split it into sentences $S_{i_1}, S_{i_2}, \cdots, S_{i_m}$. We use the publicly available *T5-small model*[1], pre-fine-tuned on SQuAD1.1, to generate possible answer spans given a sentence, to obtain answers a_j from sentences S_{i_j}.

Question Generation The question generation of *SEDAN* uses another publicly available pre-trained T5-small model[2], which is trained for question generation on SQuAD1.1 dataset, using the method described in Chan and Fan [17]. Given the generated answer a_j as above, sentence S_{i_j} along with the corresponding context passage C_i, the model generates a potentially relevant question q_j based on the probability $p(q_j \mid a_j, S_{i_j}, C_i)$ using a beam search mechanism.

In our framework, to generate the synthetic dataset, we first concatenate all the context passages from the test dataset to create the domain-specific document D. From the chunked contexts, $C_1, C_2, \cdots, C_n$ (from D), we obtain the synthetically generated QA pairs forming the training data for domain adaptation of *SEDAN*.

7.2.2 Fine-tuning Transformers for AS2

Answer Sentence Selection (AS2) involves both semantic and syntactic information to establish what information the question seeks, as well as whether a candidate sentence fulfils the requirement. Transformers [130] are a powerful architecture to capture syntactic and semantic relationships between words in natural languages. Pre-trained transformers such as BERT [22], RoBERTa[72] or DistilBERT [107] have shown to be effective in learning language models, and have been used for AS2. Prior work [37, 58] concatenated the question and the answer sentence tokens to learn their joint representation, which is then fed to a multi-layer perceptron for classifying if a sentence is answer-sentence or not. Here, the common practice is to use the [CLS] token embedding to derive sentence representations – shown to lead to sub-optimal sentence representation [103]. Hence, following the architecture of Reimers and Gurevych [103] as discussed in Section 2.5.2, we use a Siamese network to fine-tune pre-trained sentence representations to understand the contextual similarity between questions and relevant answer sentences. To this end, *SEDAN* utilises the SBERT-BERT-BASE-NLI[3] model as the base for the *two-stage* fine-tuning.

7.2.2.1 Question-Sentence Semantic Understanding:

The initial fine-tuning of the transformer model adapts the architecture towards generic AS2 systems and trains it to understand semantic similarity-based relevance between questions and corresponding answer sentences. In this respect, we use the SQuAD2.0 dataset, wherein given a question (Q), the related context (with sentences S_i), and the corresponding answer a, we mark the sentences that contain the answer text as *positive samples* while the other sentences are considered as *negative points*. Thus, a question-sentence pair (Q, S_i) is marked as 1 if $a \in S_i$, or else considered as 0.

1 https://huggingface.co/valhalla/t5-small-qa-qg-hl
2 https://huggingface.co/valhalla/t5-small-qg-hl
3 https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

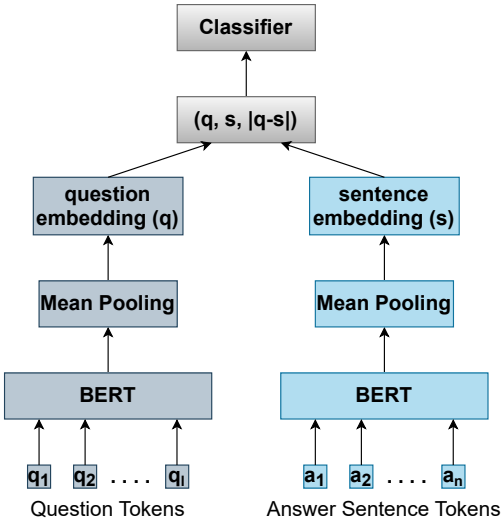

Figure 7.2: Siamese Network architecture for Transformer fine-tuning in *SEDAN*.

The setup described above tunes the transformer architecture based on learning via a Siamese network. The BERT model takes as input the text tokens and outputs their contextual embeddings, which are then mean-pooled to obtain the textual representation. We compute the embeddings for questions and sentences (within the context). The question encoding $\mathbf{q}$ and sentence embedding $\mathbf{s}$ (for the question-sentence pairs obtained above) are then concatenated along with their difference $|\mathbf{q}-\mathbf{s}|$. This question-sentence representation along with its label (whether the answer is present in the sentence) is then provided as input to a shallow classifier network to learn to identify potential answer sentences for a question. This fine-tuning network is shown in Figure 7.2. It is represented as:

$$\mathbf{q} = \text{BERT}(q) \tag{7.1}$$

$$\mathbf{s} = \text{BERT}(s), \tag{7.2}$$

$$r = concat(\mathbf{q}, \mathbf{s}, |\mathbf{q}-\mathbf{s}|) \tag{7.3}$$

$$p(s) = softmax(r\mathbf{W}) \tag{7.4}$$

where $\mathbf{W}$ is a learnable weight matrix. Observe, our training of the Siamese network is similar to the fine-tuning setting of S-BERT [103] on the NLI corpus [15], and captures semantic similarity between the questions and answer sentences.

7.2.2.2 *Self-Learning for Domain Understanding:*

The final fine-tuning of *SEDAN* involves training it to understand domain-specific contexts and relationships, as observed in Table 7.1. Here, we train *SEDAN* using the same Siamese network setting as discussed above, albeit with the synthetically generated domain-specific dataset using a pre-trained T5 model (as mentioned in Section 7.2.1).

Similar to the first-stage fine-tuning, for the generated QA pairs from the document, sentences that contain the generated answers are considered as *positive samples*, while other sentences from the document are considered as *negative samples*. The transformer model of *SEDAN* is now fine-tuned on these generated domain question-sentence pairs – enabling *domain adaptation via self-learning*.

Dataset	Generated Train Data			Test Dataset		
	#Q	#C	#QA Pairs	#Q	#C	#QA Pairs
BioASQ	3,914	422	39,750	141	451	4,564
TextbookQA	2,787	361	12,544	448	117	22,141
DROP	766	81	8,135	406	85	4,682
HotpotQA	40,841	4,979	901,416	5,901	5,860	94,646

Table 7.2: Characteristics of the *synthetically generated* training datasets. BioASQ and TextbookQA are *closed-domain* data, while DROP and HotpotQA are *open-domain*.

Thus, the overall training of our proposed *SEDAN* framework, as shown in Figure 7.1, involves (a) Siamese-network-based fine-tuning transformer architecture to generic QA scenarios, (b) synthetic generation of domain-specific training data, and (c) domain adaptation via self-learning – providing an efficient methodology to identify potential answer sentences for domain-specific user questions.

7.2.2.3 Answer Sentence Selection (AS2):

On arrival of a user question, the associated context is first split into sentences. Question-sentence pairs are created and their concatenated representations are obtained from the fine-tuned transformer of the *SEDAN* framework. The representations are then fed to the trained Siamese network for classification, to identify potential sentences that are relevant to the posed user question. The sentence(s) may be returned as an answer snippet or provided to downstream EQA for exact answer extraction.

7.3 EXPERIMENTAL SETUP

This section describes our empirical setup for evaluating *SEDAN* against existing baseline methods. We introduce the datasets, the baselines compared with, and the implementation details.

7.3.1 Datasets

As discussed previously, *SEDAN* uses a two-step fine-tuning process for the domain-aware AS2 task. In the first step, we adapt the pre-trained Sentence-BERT model to the QA task using SQuAD2.0 dataset [102], containing 129,353 unique questions and a total of 334,364 QA sentence pairs. *SEDAN* utilises the synthetically generated domain-grounded dataset from the T5 module as detailed in Sec. 7.2.1 for adapting the model to a specific domain. The datasets are obtained from `https://huggingface.co/datasets/mrqa` [30], and their details are shown in Tab. 7.2.

- **BioASQ** [87]: BioASQ involves information extraction on biomedical semantic indexing and QA pairs annotated by domain experts. We take 70% of the dataset as training and 30% as the test set. The test split contains 141 unique questions, 451 unique contexts and 4,564 QA sentence pairs. The synthetic dataset contains 3,914 and 422 unique questions and contexts respectively, with 39,750 QA sentence pairs.
- **TextbookQA** [49]: This dataset contains QA pairs from middle school science curricula. We split the dataset into 70% for training and 30% for testing. The test dataset contains

448 and 117 unique questions and contexts respectively, with 4,682 QA sentence pairs. The synthetic dataset contains 2,787 and 361 unique questions and contexts, comprising of 12,544 QA sentence pairs.

- **DROP** [25]: The DROP dataset involves quantitative reasoning questions over Wikipedia paragraphs. The original dataset contains numeric answers to be inferred from the context, however, we only consider those questions where the answer is explicitly mentioned in the passage text. The test dataset contains 406 questions, 85 passages, and 4,682 QA sentence pairs. The generated synthetic dataset contains 766 unique questions over 81 passages 8,135 QA sentence pairs.
- **HotpotQA** [153]: HotpotQA is a multihop QA dataset to answer questions over Wikipedia. The test contains 5,901 unique questions, 5,860 unique contexts and 94,646 QA sentence pairs. The synthetic data contains 40,841 unique questions, 4,979 contexts with 901,416 QA sentence pairs.

BioASQ and TextbookQA are *closed-domain*, while DROP and HotpotQA are *open-domain* – showcasing the performance of *SEDAN* in AS2 for diverse settings. For the above datasets, we evaluate if the sentence(s) retrieved by an AS2 model contains the ground truth answer text span within them.

The original questions in the test set are not visible to *SEDAN* during the domain adaptation process, the synthetic QA pairs (used during self-learning-based training) are generated from the test documents. During the training process, the synthetic datasets are created solely from the articles present in the test set, without any utilization of the training dataset.

7.3.2 Baseline Approaches

We compare our proposed approach against several state-of-the-art AS2 models as well as passage ranking models as baselines.

- **Dual-CTX** [58] is a RoBERTa [72] based state-of-the-art AS2 model, utilising global article context along with local passage context. It was also shown that using the Transfer AND Adapt (TANDA) approach, based on an initial fine-tuning on the ASNQ dataset, improves the performance of AS2 on other data (referred to here as Dual-CTX-TA). In the TANDA approach, the Dual-CTX framework is first trained on the ASNQ dataset to transfer the RoBERTa model for the AS2 task. In the second stage, the models trained in the first step are fine-tuned on a domain-specific dataset to achieve domain-adaptation.
- **Reranker** [36] is a lightweight and efficient approach based on language model based reranking for information retrieval (IR), QA and other natural language processing tasks. The model uses localised contrastive estimation loss to finetune a transformer model for text reranking.
- **Propagate-Selector** [155] utilises graph structures for detecting candidate answer sentences. It constructs a graph using sentences in the passage as nodes, with edges between sentences and questions, for utilising GNN for AS2.
- **TF-Ranking** [96] is a library for Learning-to-Rank (LTR) framework for IR tasks using the TensorFlow platform.

Model	BioASQ					TextbookQA				
	MRR	MAP	R@1	R@3	R@5	MRR	MAP	R@1	R@3	R@5
Dual-CTX	68.78	62.06	55.53	59.55	70.58	18.91	14.17	8.93	8.77	11.52
Dual-CTX-TA	71.68	64.83	59.95	63.79	72.03	21.16	16.17	10.26	11.75	15.90
Propagate-Selector	45.26	47.48	26.99	48.16	66.54	11.48	12.06	4.02	8.22	12.74
Reranker	72.51	65.49	59.29	64.63	74.42	55.13	42.21	43.75	40.77	**48.49**
TF-Ranking	54.92	50.92	36.06	49.19	64.59	16.17	13.08	6.03	7.51	11.43
BM25	71.30	64.59	56.19	64.89	75.56	52.23	32.69	39.06	42.63	45.79
SEDAN	**78.07**	**70.04**	**65.26***†	**72.16***†	**80.26***†	**57.26**	**43.57**	**44.86**†	**45.02**†	48.04†

Model	DROP					HotpotQA				
	MRR	MAP	R@1	R@3	R@5	MRR	MAP	R@1	R@3	R@5
Dual-CTX	39.92	38.02	13.15	49.35	60.89	61.37	57.10	47.43	59.67	69.32
Dual-CTX-TA	59.43	55.52	41.90	54.89	79.40	**67.33**	**64.44**	**53.00**	**69.23**	**79.17**
Propagate-Selector	40.40	41.23	17.41	41.90	75.15	42.11	44.33	24.06	46.85	66.68
Reranker	56.53	54.53	36.43	62.21	76.57	61.44	59.54	43.87	64.76	77.75
TF-Ranking	42.41	40.68	21.86	40.69	67.56	33.29	31.90	16.11	27.70	42.84
BM25	65.24	63.25	49.79	66.83	79.25	47.49	44.86	27.89	47.22	64.52
SEDAN	**73.79**	**70.71**	**59.71***†	**76.51***†	**86.41***†	64.19	61.82	48.11†	66.01†	78.64†

* refers to statistically significant result for SEDAN compared to Dual-CTX-TA, Reranker and BM25 with $p < 0.1$, while † refers to statistically significant result for SEDAN compared to Propagate-Selector and TF-Ranking with $p < 0.01$.

Table 7.3: Performance of the algorithms trained using synthetically generated dataset on different test dataset for the AS2 task.

- **BM25** [6] is a traditional IR model based on BoW model and uses term frequency and inverse document frequency to rank answer sentences given a question.

7.3.2.0.1 Evaluation Measures:

We evaluate the different AS2 models based on the following:

- *Mean Reciprocal Rank (MRR)* – It is defined as the average harmonic mean (over all questions) based on the rank of the first correct answer sentence reported, given a question. Mathematically, $MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_{s_i}}$, where $rank_{s_i}$ denotes the rank of the first correct answer sentence returned for question Q_i, and $|Q|$ denotes the total number of questions.

- *Mean Average Precision (MAP)* – It computes the mean (over all questions) of the average precision in identifying the answer sentences of a question. Thus, $MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|} AP_i$, where $AP_i = \frac{1}{K_i}\sum_{j=1}^{K_i} P@j$ computes the accuracy of the answer (for query Q_i) being present within the top-j sentences returned (i.e., P@j), and K_i denotes the total number of answer sentences.

- *Recall@k (R@k)* – It measures the fraction of the total number of correct answer sentences (within the context) retrieved in the top-k sentences by the methodologies. We report for $k = \{1, 3, 5\}$. Mathematically, we have,

$$R@k = \frac{\text{\# correct answer sentences retrieved}}{\min(\text{\# total answer sentences}, k)}$$

For the Extractive Question Answering (EQA) use-case application task, we measure the performance of the models on Exact Match (EM) and *F1* metrics as defined in Section 6.3.3.

For a fair comparison, we train the *Reranker* with the BERT-base model. As the source code of Dual-CTX is unavailable online, we evaluate its performance on its best effort re-implementation

with the RoBERTa-base model[4]. As the focus of this work is to alleviate the need for annotated domain-specific training data, and hence we do not consider any document or paragraph-level information that might not be available. Hence, such information is not provided to the Propagate-Selector, and individual passages are used to construct the global context for Dual-CTX.

We also evaluate the impact of the baseline AS2 techniques on the downstream EQA task. For this, we consider the state-of-the-art *SpanBERT* model [45], a BERT model fine-tuned using span masking for predicting precise answer spans robustly. We use their publicly available model[5] fine-tuned on the SQuAD2.0 dataset.

7.3.3 Implementation Details

We use spaCy [43] to split the provided contexts into sentences during the synthetic dataset construction via T5 model (Section 7.2.1). K_{GQA} determines the number of sentences in the context used by the T5 model for QA generation. We set $K_{GQA} = 10$ for BioASQ, DROP and HotpotQA, while for TextbookQA it is set to 41 – based on the average length of the test documents in each dataset.

For the AS2 task, we train the transformer models in *SEDAN*, Reranker and Dual-CTX with a batch size of 256 on 2 Tesla V100 GPUs, while for the TF-Ranking model and Propagate-Selector we use batch sizes of 256 and 16 respectively, on NVIDIA 1080Ti GeForce GPUs. We optimise the models with the Adam optimiser [51] with a learning rate of 10^{-6} and a linear warm-up over 10% of the training data. We set the weight decay parameter to 0.01 for regularisation. We train all models for 50 epochs on the BioASQ and DROP datasets, and for 10 epochs on HotpotQA and TextbookQA dataset. The hyperparameter values were set using grid-search methodology.

For evaluating on the EQA task, we prune the context passage to contain the top $K_{pruning}$ candidate answer sentences (based on AS2 classification scores) and the candidate sentences are concatenated and presented to the SpanBERT EQA model as context for the question posed. We set $K_{pruning}$ to min (50% of the number of document sentences, K_{min}). We set K_{min} for BioASQ, TextbookQA and DROP datasets to 4, 6 and 6 respectively.

7.4 RESULTS AND DISCUSSION

This section reports the performance results obtained by *SEDAN* along with the other methods for AS2 and EQA tasks on different *closed-* and *open-domain* datasets. We also analyse the semantic relationship between question and answer sentence embeddings learnt by *SEDAN*, via-an ablation study. For completeness, we explore the performance of the methodologies in settings when domain-specific training data might be available.

4 Our re-implementation produced similar results on SQuAD as reported by the authors.
5 https://huggingface.co/mrm8488/spanbert-base-finetuned-squadv2

Model	BioASQ		TextbookQA		DROP	
	EM	F1	EM	F1	EM	F1
Full Context	39.60	49.62	27.90	35.87	23.07	28.49
Dual-CTX-TA	36.94	46.98	15.18	20.64	23.68	**29.82**
Reranker	39.38	49.39	25.89	33.85	21.65	27.72
TF-Ranking	30.75	38.53	16.74	21.78	18.62	23.73
BM25	36.94	47.09	27.23	34.99	20.04	27.05
SEDAN	**40.04**	**50.94**	**28.79**	**36.33**	**23.88**	29.01

Table 7.4: Performance of AS2 models for EQA task.

Model	BioASQ					DROP				
	MRR	MAP	R@1	R@3	R@5	MRR	MAP	R@1	R@3	R@5
Domain Fine-tuning + Siamese	71.03	64.17	55.97	65.04	78.27	61.20	58.23	39.47	71.28	82.62
Domain Fine-tuning + Cosine	71.68	65.26	55.97	67.51	79.17	64.70	62.04	45.95	71.22	82.60
Full Fine-tuning + Cosine	76.31	67.83	63.71	68.65	77.94	69.20	66.81	53.64	73.00	84.02
SEDAN (Full architecture)	**78.07**	**70.04**	**65.26**	**72.16**	**80.26**	**73.79**	**70.71**	**59.71**	**76.51**	**86.41**

Table 7.5: Ablation study on fine-tuning and classification modules in *SEDAN*.

7.4.1 Performance of *SEDAN* on AS2 Task

Table 7.3 reports the performance of *SEDAN* along with other competing approaches trained using the synthetically generated dataset. For *closed-domain* or domain-specific datasets like BioASQ and TextbookQA, we observe that *SEDAN* outperforms all other baselines. *SEDAN* achieves a performance improvement of around **5%** on MRR, MAP and R@1 measures over the state-of-the-art Dual-CTX-TA, and Reranker approaches, while Propagate-Selector and TF-Ranking report the lowest performances.

On the open-domain DROP dataset, we also achieve the best performance with a significant improvement over the baselines. This can be attributed to a better understanding of the Question-Answer sentence semantic relationship in *SEDAN* across different regions of the document. Dual-CTX-TA performs best on HotpotQA, a multi-hop QA dataset, as it captures the global context using bag-of-words, thereby capturing the inter-sentence information. Despite not incorporating global context, the *SEDAN* model achieves high performance and is the second best-performing model. This can be attributed to its ability to effectively capture semantic relatedness between terminologies and sentences within the document.

We conduct a one-sided binomial test to study the statistical significance of the reported results on Recall@k (R@k) metrics. As most of the questions across the datasets contain only one correct answer sentence, we treat the R@k values as Bernoulli variables. We consider the null hypothesis, H_0 : *The performance of method X is comparable to that of SEDAN*; where X∈{Dual-CTX-TA, Reranker, BM25} – the top-3 best-performing approaches. In Table 7.3, we find that the performance of *SEDAN* is significantly better than the baselines, thereby rejecting the above null hypothesis with a p-value $p < 0.1$. That is, the performance of *SEDAN* is observed to be *significantly better* than the existing methodologies in both *open- and closed-domain* datasets. Since the sample test size is quite low for the datasets (see Table 7.2), we consider a confidence threshold of 90% (instead of the standard 95%) for significance testing, as noted in Leamer [59] for data with fewer samples.

It should be noted that all models are fine-tuned on the domain-specific data synthetically generated as in *SEDAN*. We later show in Section 7.4.3, that without such fine-tuning, the embed-

Model	BioASQ					TextbookQA					DROP				
	MRR	MAP	R@1	R@3	R@5	MRR	MAP	R@1	R@3	R@5	MRR	MAP	R@1	R@3	R@5
Dual-CTX-TA	**91.10**	**88.91**	**85.39**	**91.85**	**95.28**	21.19	16.27	10.26	12.57	16.25	65.80	62.54	49.59	67.84	80.45
Reranker	84.42	78.07	74.55	79.68	88.27	54.22	41.70	41.96	42.18	48.00	73.59	70.73	59.31	76.31	84.00
BM25	71.30	64.59	56.19	64.89	75.56	52.23	39.69	39.06	42.63	45.79	65.24	63.25	49.79	66.83	79.25
SEDAN	89.21	85.00	83.18	86.20	92.12	**57.73**	**44.97**	**45.75**	**44.86**	**50.64**	**75.79**	**73.17**	**61.94**	**79.14**	**89.03**

Table 7.6: Performance of baseline methods in presence of gold training data.

	Text
Q	What disease is characterised by an abnormal production of blood cells?
Pred.	It is characterised by an abnormal production of blood cells, usually white blood cells.
Gold	Leukemia is a cancer of the blood or bone marrow.
Q	Which domain of TIA-1 is necessary for stress granule assembly?
Pred.	The PRD of TIA-1 exhibits many characteristics of prions: concentration-dependent aggregation that is inhibited by the molecular chaperone heat shock protein (HSP)70; resistance to protease digestion; sequestration of HSP27, HSP40, and HSP70; and induction of HSP70, a feedback regulator of PRD disaggregation.
Gold	The RNA recognition motifs of TIA-1 are linked to a glutamine-rich prion-related domain (PRD).

Table 7.7: Sentence Retrieval Analysis of *SEDAN*

dings are not rich enough to capture domain-specific semantic information. In the remaining experiments, we ignore the two lowest-performing approaches, Dual-CTX and Propagate-Selector.

Overall, we find that *SEDAN* captures both closed-domain and open-domain contextual information that can identify answer sentences effectively – and provides an effective self-learning based AS2 framework in diverse settings.

7.4.2 Performance on Downstream EQA Task

Although AS2 has been positioned to benefit downstream QA (by providing an enriched context for exact answer text extraction), its actual impact on EQA has not been studied in the literature. Here, we showcase how providing the top-k candidate answer sentences (from AS2) might impact the performance of EQA. We analyze the impact of AS2 on the performance of downstream EQA task as a use-case application, and hence compare the impact of different AS2 approaches (as opposed to other EQA specific trained models).

Table 7.4 reports the performance of the different models on the EQA task for BioASQ, TextbookQA and DROP datasets. It can be observed that *SEDAN* outperforms other existing baselines in both EM and F1 scores across the closed- and open-domain datasets. Interestingly, we observe that all the other baselines perform worse than the *Full Context* (for domain-based QA), where the entire original passage is provided to the EQA module without any AS2 processing. Thus, although existing AS2 models accurately retrieve potential answer sentences, they may not provide a good context for enhancing the EQA system. However, the context obtained from *SEDAN* provides better contextual and semantic relevant information for a question, thereby improving the performance of EQA – by learning-rich semantic information between questions and sentences. To the best of our knowledge, *SEDAN* is the first framework to show performance improvements for AS2, and also positively impact downstream Extractive Question Answering (EQA) task.

7.4.3 Ablation Study of *SEDAN* Modules

The *SEDAN* framework involves *two-stage fine-tuning* and *Siamese network* based representation learning. Here we perform a small-scale ablation study to understand the impact of the individual components on the BioASQ and DROP dataset.

We initially consider a generic Sentence-Transformer architecture with no domain fine-tuning and extracting the sentences in the context that demonstrate the highest cosine similarity with the question. As per expectation, this variation does not perform well on the domain-specific datasets, since it has no knowledge of semantic understanding within the domain. We next remove the first-stage fine-tuning in *SEDAN* , i.e., we do not pre-align it for QA, and evaluate using both the Siamese network classification, as discussed in Section 7.2.2.1, and the question-sentence cosine similarity score. We observe that simple single-stage domain adaptation, based on the synthetically generated QA pairs, improves AS2 performance on the closed-domain dataset. Finally, we perform the full two-step fine-tuning, but use the cosine similarity to identify candidate answer sentences, instead of the Siamese network classification. Similar to the TANDA approach [58], interestingly, the complete fine-tuning leads to an improvement in performance over the other variations, and together with the Siamese network sentence extraction classifier, i.e., the proposed *SEDAN* framework, provides the best results across all the ablation study scenarios (Table 7.5).

7.4.4 Performance with Gold Training Data

For completeness, we also study the behaviour of the approaches in scenarios where annotated training data might be available. From Table 7.6, we observe that *SEDAN* outperforms the baselines on TextbookQA and DROP, while reporting the second-best (comparable) on BioASQ. In fact, we see only a slight improvement in the performance of *SEDAN*, in the presence of annotated data. On the other hand, the remaining baselines showcase a marked performance enhancement (compared to the unsupervised setting). This indicates that existing methodologies are primarily geared toward expensive training procedures for acceptable performance. Thus, we see that *SEDAN* is geared towards domain-specific self-learning and is robust to diverse scenarios – effective for supervised and unsupervised AS2.

7.4.5 Error Analysis

This section highlights a set of interesting scenarios underlying the working of *SEDAN*. Table 7.7 shows two examples where *SEDAN* predicts the correct answer sentence (i.e., containing the answer span), but is different from the gold answer sentence.

In the first example, we can observe that although different from the annotated example, the model selects the sentence with a very high contextual overlap compared to the annotated sample. Here, the pronoun "It" in the predicted answer sentence from *SEDAN*, refers to the correct answer (when read with the full document context). It is worth noticing that a co-reference resolution stage can address the issue to enable the downstream EQA task to choose the correct answer span.

In the second example, we find that both the predicted sentence and the annotated ground-truth sentence, although different, contain the correct answer text – *PRD*. It can be observed

that *SEDAN* relates the word *disaggregation* in the predicted sentence to the word *assembly* in the question text to come up with the answer sentence. The ground truth sentence contains the full form of the acronym *PRD* and hence might be a bit more helpful in answering the question, in terms of user experience.

7.4.6 Obstacles to Deployment

Model Size. The proposed *SEDAN* framework utilises a pre-trained sentence-transformer model and two T5 models. Hence, the proper training and deployment of our system depends on sufficient computational resources.

Domain Data Update. Although *SEDAN* is self-supervised, it needs to be trained individually for each application domain of usage. A single model spanning multiple domains might suffer from degraded performance due to "information collusion" across the domains. The presence of such multiple large models might impede certain deployment opportunities. Further, the models need to be re-trained with new emerging data within the domains.

Impact of Synthetic Data Quality. The performance of *SEDAN* depends on self-supervised domain adaptation based on the synthetically generated training question-answer pairs. In the case of niche domains (e.g., molecular geometry, etc.), using such pre-trained language models (which are not pre-trained on such domain data) might not provide quality synthetically generated training data. This might lead to a sub-par performance of our proposed system and might hinder the adoption of such applications.

7.5 SUMMARY

This chapter explores the self-supervised training of the *unstructured knowledge selector* shown in Figure 1.2, with a focus on retrieving relevant information from unstructured sources like domain-specific documents or articles to provide answers to user queries in a QA scenario.

First, we propose *QASAR*, a novel and efficient EQA framework for *self-learning* based *domain adaptation* on closed-domain applications. We showcase how synthetically generated domain-specific training data using large language models can enable fine-tuning of existing "generic" EQA models for improving domain-specific EQA platforms – eliminating the need for any expensive annotated data creation. We also propose a novel context retrieval methodology based on identifying semantically relevant contexts via sentence embeddings. Extensive experiments of diverse open- and closed-domain QA datasets, showcase statistically significant performance improvements compared to existing techniques. We also depict the robustness and scalability of our *QASAR* framework to diverse domains and empirical settings – enabling wider adoption within chatbots in an enterprise setting.

Addressing the task of AS2, we proposed *SEDAN*, a framework for *self-learning* based *domain adaptation* for efficient Answer Sentence Selection (AS2). We showcase how a two-stage fine-tuning process using synthetically generated domain data, along with a Siamese network-based classification, enables a better understanding of the question-answer sentence semantic relationships. We depict results on closed- and open-domain datasets and exhibit improved accuracy in answer sentence identification, as well as for downstream tasks such as EQA. We demonstrated

the advantages of employing synthetic datasets for adapting the EQA and AS2 components of a chatbot to a specific domain. These two components can collaborate to handle domain-specific user queries. The domain-grounded AS2 component can aid in selecting the relevant domain-specific EQA component for answer extraction. The extracted answers can facilitate a chatbot in providing responses to domain-specific queries effortlessly. A limitation of both works is the need to analyze the impact of synthetic dataset size on performance. Additionally, the performance of the systems when working together should be examined.

8 RESPONSE GENERATION FROM STRUCTURED AND UNSTRUCTURED KNOWLEDGE SOURCES

Generating informative, coherent and fluent responses to the dialogue context is challenging yet critical for a rich user experience and the eventual success of chatbot. Knowledge-grounded chatbot leverage external knowledge to induce relevant facts in a dialogue. These systems need to understand the semantic relatedness between the dialogue context and the available knowledge, thereby utilising this information for response generation. Although various innovative models have been proposed, they neither utilise the semantic entailment between the dialogue history and the knowledge nor effectively process knowledge from both structured and unstructured sources. In this chapter, we propose PICKD, a two-stage framework for knowledgeable dialogue. The first stage involves the *Knowledge Selector* (see Figure 1.2) choosing knowledge pertinent to the dialogue context from both structured and unstructured knowledge sources. PICKD leverages novel *In-Situ* prompt tuning for knowledge selection, wherein prompt tokens are injected into the dialogue-knowledge text tokens during knowledge retrieval. The second stage employs the *Response Generator* (Figure 1.2 for generating fluent and factual responses by utilising the retrieved knowledge and the dialogue context. Extensive experiments on three domain-specific datasets exhibit the effectiveness of PICKD over other baseline methodologies for knowledge-grounded dialogue. The source code is available at `https://github.com/rajbsk/pickd`.

8.1 INTRODUCTION

With the proliferation of personal assistants (Siri, Alexa, etc.), research on chatbot systems has gained much traction. Inducing relevant information in responses leads to a fluent, engaging and coherent conversation with a dialogue system. While language models help to develop fluent chatbot systems [166, 169], such systems lack the necessary tools for generating accurate responses. Researchers have utilised external knowledge from either unstructured knowledge sources such as Wikipedia articles [23, 151], domain-grounded documents [161] or structured sources like KGs [170, 176] in assisting dialogue agents generate informative responses.

Knowledge-grounded chatbot systems must interpret the semantic relatedness between the dialogue context and the external knowledge. For example, in Figure 8.1, while responding to the utterance "Is London heavily populated?", the model needs to understand the context encompasses "London" and then utilise the knowledge triple "London; Population; 7.1 million" when generating the response text. Similarly, when responding to the utterance "Yes, London Bridge is a good monument there. Anything else?", the agent needs to register the contextual overlap between the dialogue history and the knowledge sentence "London Bridge and the

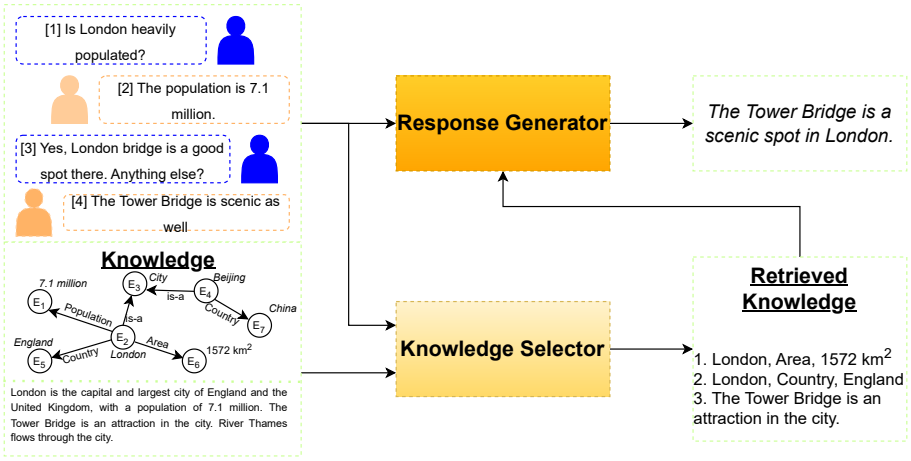

Figure 8.1: Modular overview of the PICKD framework. The *Knowledge Selector* retrieves dialogue context-relevant knowledge from the structured and unstructured knowledge available. The retrieved knowledge and along with the dialogue context is then sent to the *Response Generator* for producing informative and fluent responses.

Bridge Tower are major attractions in London.". It should then leverage this information while generating utterance 4.

While prior works have posited novel frameworks to address this task, they tend to cater exclusively to unstructured knowledge sources like paragraphs/documents [23] or solely to structured sources like KGs [84]. Zhou et al. [172] proposed the KdConv dataset: a knowledge-grounded dialogue spanning three domains. Additionally, they suggested adapting Seq2Seq and the Hierarchical Recurrent Encoder-Decoder Framework with knowledge memory for utilising both structured and unstructured sources of knowledge during response generation. Wang et al. [133] improved upon this work with their innovative Relation-Transition aware Knowledge Grounded Dialogue (RT-KGD) framework for informative response generation. RT-KGD uses a heterogeneous graph transformer network to capture the information flow of the dialogue and leverages the BART [62] model for response generation. While the methodologies are novel in their own right, the models do not represent the dialogue context and the knowledge elements in the same semantic space. As a result, the models tend to generate irrelevant facts in the response.

Prompting and prompt-tuning methodologies [61, 168] has shown to be a powerful tool in classification tasks without the need of tuning the parameters of a language model. Prompt-tuning is a resource-efficient approach for fine-tuning language models for specific tasks, leading to lower resource requirements and improved parameter efficiency. Towards this, we explore prompting methodologies for the knowledge-grounded conversation task. In this chapter, we propose PICKD, a novel knowledgeable dialogue generation framework employing *In-Situ* prompt tuning for knowledge selection. PICKD initially trains the *Knowledge Selector* with the *In-Situ* prompt tuning paradigm for context-relevant knowledge retrieval. PICKD uses the RoBERTa [72] architecture as the base model for the *Knowledge Selector*. During *In-Situ* prompt tuning, we inject prompt tokens within the dialogue context and knowledge elements and then fine-tune the prompt tokens along with the classification heads during training. During this tuning procedure, the prompt tokens learn the semantic relevance between the dialogue context and the knowledge elements. Thus, the *Knowledge Selector* learns to retrieve knowledge from

structured and unstructured sources compatible with the dialogue context. The retrieved knowledge along with the dialogue context is then sent to the *Response Generator* for knowledgeable dialogue generation. The *Response Generator* utilises the BART architecture for response generation and is fine-tuned to produce a knowledgeable and coherent response. In summary, our contributions are as follows:

- PICKD, a two-stage framework for Knowledge Grounded Dialogue (KGD) grounded on both structured and unstructured sources of knowledge.
- Novel *In-Situ* prompt tuning paradigm for *Knowledge Selection* in domain-specific knowledge-grounded dialogue for retrieving context-relevant knowledge by enabling the learning of semantic relationships between the context and knowledge elements.
- Extensive evaluations on multiple domain-specific knowledge-grounded dialogue datasets with ablation variants to demonstrate the strong evidential improvements of PICKD over other baselines for knowledge-grounded dialogue.

8.2 METHODOLOGY

We begin with formally defining the problem statement. Thereafter, we introduce the *Knowledge Selector* module, our novel *In-Situ* prompt tuning framework for knowledge selection. Finally, we describe the workings of the *Response Generator* for knowledgeable dialogue generation.

8.2.1 Formal Problem Definition

We are given the dialogue context $C = \{u_1, u_2, u_3, ..., u_{n-1}\}$, where u_i is an utterance from the conversation history, triples $\{(h_1, r_1, t_1), (h_2, r_2, t_2), ..., (h_{|K_k|}, r_{|K_k|}, t_{|K_k|})\}$ from a KG, where K_k is the number of triples in the KG and descriptive unstructured text set $S = \{S_1, S_2, ..., S_{S_k}\}$, where S_k is the number of articles in the set and each article comprises of multiple text sentences. The objective is to generate a coherent response u_n that is not only grammatically correct but also informative, leveraging the context and the knowledge sources available.

8.2.2 Contextual Prompting for Knowledge Selection

The *Knowledge Selector* module of PICKD retrieves knowledge from structured and unstructured sources to be leveraged by the *Response Generator*. The *Knowledge Selector* module is trained to understand the semantic congruence between the dialogue context and knowledge elements using the novel *In-Situ* prompt tuning paradigm. The knowledge facts from structured and unstructured sources are then ranked according to their relevance by the module, and appropriate knowledge is then sent to the *Response Generator*.

8.2.2.1 Prompting Architecture.

The *Knowledge Selector* module of PICKD employs a pre-trained RoBERTa model as its backbone. The RoBERTa[1] model takes the dialogue context C and the knowledge K as inputs. The knowl-

1 https://huggingface.co/hfl/chinese-roberta-wwm-ext

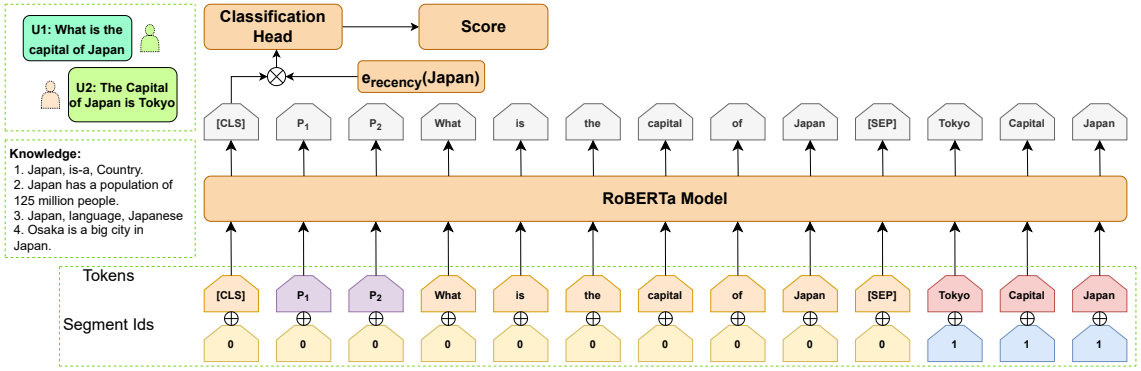

Figure 8.2: Overview of the *In-Situ* prompting tuning framework. The module uses the RoBERTa architecture as its base model. The input token embeddings are supplemented with segment token embeddings. The output of the [CLS] token embedding is then sent to a classification layer along with the recency embedding of the head entity of the knowledge element. This classification layer learns to assign higher scores to relevant knowledge. During the training phase, the prompt tokens along with the classification head and the recency embeddings are learnt by the model. Here, $\oplus$ denotes addition, while $\otimes$ denotes the concatenation operation.

edge K comprises a triple of an entity h from the KG, a relation r from the KG, and a tail entity t from the KG or a sentence from the paragraph description of h. As unstructured knowledge is available in paragraph form, it is split into individual sentences. To ensure the input format of knowledge from KG and paragraphs are uniform, the input t can be either the tail entity or a sentence about h from its paragraph description. The input from K to the RoBERTa is the strings of h, r and t concatenated together. For example, Figure 8.2 showcases an example scenario wherein the *knowledge selector* needs to retrieve knowledge about *"Japan"* relevant to the conversation context. The input K from the KG triple *("Japan, is-a, Country")* takes the form *"Japan is-a Country"*. On the other hand, the input K from the unstructured text *"Osaka is a big city in Japan."* takes the form *"Japan Information Osaka is a big city in Japan."*. The concatenation of context C and knowledge K form the initial RoBERTa input. PICKD injects prompt tokens into the input and adds segment embeddings to the resulting input tokens. This enables the capture of semantic dependencies between the context and knowledge elements. The token embeddings then pass through a RoBERTa layer, and the final representation of the [CLS] token along with the recency embedding of h is sent to the classification layer for scoring. Recency embeddings are learnable vectors that capture the relevance of h in the dialogue history, similar to positional embeddings. This ensures that head entities mentioned recently have more influence during response generation. Formally, we define the setup as follows:

$$P_{prompt} = P_1, P_2, ..., P_{k_{prompt}} \tag{8.1}$$

$$D_{dial} = D_1 D_2 ... D_n \tag{8.2}$$

$$K_{know} = K_1 K_2 ... K_k \tag{8.3}$$

$$T_{input} = [CLS] P_{prompt} D_{dial} [SEP] K_{know} \tag{8.4}$$

$$h_{CLS}, h_{p_1}, ..., h_{K_k} = \text{RoBERTa}(T_{input}) \tag{8.5}$$

$$v = \text{MLP}(h_{CLS}; \mathbf{e}_{recency}) \tag{8.6}$$

$$\text{score} = \text{Softmax}(v) \tag{8.7}$$

where P_i, D_j and K_k denote the i^{th} prompt token, j^{th} dialogue history token and the k^{th} knowledge tokens. The input to the RoBERTa model is composed of the prompt tokens, dialogue tokens and knowledge tokens concatenated together. A [CLS] token is added to the beginning of the sequence denoting start of the sequence and a [SEP] token is used to separate the prompt and dialogue tokens from the knowledge tokens as shown in Equation 8.4. This input is sent through a RoBERTa layer as shown in Equation 8.5 wherein segment embeddings are added to the token embeddings before realising the contextual embeddings. PICKD then concatenates the RoBERTa representation of [CLS] token with the head entity recency embedding and is then sent through a classification MLP layer as detailed in Equation 8.6 and 8.7. The model is trained by minimising the cross-entropy loss. During inference, the knowledge elements are ranked following Equation 8.7. The top-k elements are then sent to the *Response Generator* for response generation.

8.2.3 BART fine-tuning for Response Generation

The *Response Generator* is trained to utilise the semantic information from the dialogue context and the retrieved knowledge to generate the knowledgeable response. Similar to Wang et al. [133], PICKD employs the BART architecture[2] [62] for response generation.

More formally, the input of the BART model consists of the dialogue context and the knowledge retrieved. At the N^{th} turn, input from the dialogue context takes the form "*[CLS]* u_1 *[SEP]* u_2 ... *[SEP]* u_{N-1} *[SEP]*", where *[CLS]* and *[SEP]* tokens are special tokens denoting the start of the sequence and sentence boundary token respectively. Knowledge input is constructed as "*[CLS]* K_1 *[SEP]* K_2 *[SEP]*... $K_{k_{know}}$ *[SEP]*", where k_{know} is the cardinality of the retrieved knowledge set and K_i is the knowledge fact defined as the concatenation of the head entity, relation and tail entity. The context concatenated with the knowledge is sent as input to the BART encoder layer as detailed in Equation 8.8.

$$\begin{aligned} h_1^C, h_2^C, h_3^C ... h_n^C = \text{BART}_{enc}([\text{CLS}]u_1[\text{SEP}]u_2 ... u_{N-1}[\text{SEP}] \\ [\text{CLS}]K_1[\text{SEP}]K_2 ... K_{k_{know}}[\text{SEP}]) \end{aligned} \tag{8.8}$$

$$G = \text{BART}_{dec}(h_1^C; h_2^C; h_3^C; ...; h_n^C) \tag{8.9}$$

$$\mathcal{L}_{decoder} = -\frac{1}{|Y|} \sum_{t=1}^{|Y|} \log(Y_t = G_t) \tag{8.10}$$

where h_i^C is the contextual representation of the i^{th} token in the input sequence and n is the length of the concatenated input. The encoded representations are then sent to a BART decoder layer for knowledge-grounded response generation as defined in Equation 8.9, where G is the response representation. The autoregressive BART_{dec} model is then trained by minimising the cross-entropy loss over the ground-truth tokens as described in Equation 8.10, where G_t is the ground-truth token and Y_t is the predicted token at timestep t.

2 https://huggingface.co/fnlp/bart-base-chinese

8.3 EXPERIMENTAL SETUP

This section details the setup and the evaluation of PICKD against the baseline methodologies on the KGD task.

8.3.1 Datasets

Table 8.1: KdConv dataset characteristics

	Film	Music	Travel
# Train dialogues	1,200	1,200	1,200
# Dev dialogues	150	150	150
# Test dialogues	150	150	150
# KG entities	7,477	4,441	1,154
# KG rels	4,939	4,169	7
# KG triples	89,618	56,438	10,973

We conduct experiments on the KdConv dataset [172] to evaluate the effectiveness of PICKD. KdConv is a knowledge-grounded Chinese dialogue dataset spanning the *film*, *music* and *travel* domains. The dialogue utterances are annotated with knowledge from either a KG or from unstructured texts. By grounding the dialogues on both structured and unstructured knowledge sources, our choice of the dataset enables the evaluation of its performance in the presence of diverse knowledge types, including both structured and unstructured knowledge. The dataset contains 1,200, 150 and 150 dialogues in the training, development and test set respectively in each domain. The dialogues in the dataset have an average of 19.0 utterances and 10.1 annotated knowledge triples. The choice of this dataset enables a robust evaluation of PICKD on multiple domains in the presence of both structured (KGs) and unstructured knowledge (articles, documents) sources. The dataset statistics are detailed in Table 8.1.

8.3.2 Baseline Methodologies

Following Wang et al. [133] and Zhou et al. [172], we consider the following methodologies as the baselines for the task:

- Seq2Seq [172]: An attention-based encoder-decoder framework trained to generate the response text conditioned on the dialogue history. This methodology does not utilise external knowledge for response generation.
- HRED [172]: A Hierarchical Recurrent Encoder-Decoder framework that employs a context RNN to inject the historical utterances into the context state. The model then generates the response conditioned on the dialogue context. This model does not utilise external knowledge as well.
- Seq2Seq + Know [172]: An encoder-decoder framework trained to generate responses conditioned on the dialogue history and the external knowledge.
- HRED + Know [172]: Similar to HRED, however, the architecture generates a response conditioned on the dialogue context and the external knowledge.

- BART [133]: An encoder-decoder framework employing the BART architecture for generating dialogue responses. Similar to Seq2Seq and HRED baselines, this model does not utilise external knowledge.

- RT-KGD [133]: A relation-transition aware knowledge grounded dialogue generation framework. RT-KGD employs Heterogeneous Graph Transformers to inject KG information into a BART model for response generation.

8.3.3 Evaluation Metrics

Automatic Metrics: Following Zhou et al. [172] and Wang et al. [133], we adopt perplexity, BLEU-N [92] and distinct-N metrics for automatic model evaluation. Perplexity evaluates the capacity of the model to generate the ground-truth response. Lower perplexity denotes better confidence in generating ground-truth responses. Blue-N measures the N-gram overlap between the generated response and the ground-truth response, while distinct-N measures the diversity of N-gram tokens in the response generated.

Manual Metrics: Following Zhou et al. [172] and Wang et al. [133], we use fluency and coherence as the metrics for human evaluation. Fluency measures whether the generated responses are grammatically correct, fluent and human-like. On the other hand, coherence measures if the generated responses are coherent to the dialogue context and consistent with the available knowledge. The generated responses are assessed by evaluators who are native speakers of the Chinese language. The evaluation is conducted using a three-point scale where the responses are assigned discrete values of 0, 1, or 2, with 0 being the worst and 2 being the best. Each annotator evaluated 50 examples of context-response pairs from each domain on the fluency and coherence metrics.

8.3.4 Implementation Details

We utilise the PyTorch and the Huggingface libraries[3] for developing PICKD. The *Knowledge Selector* is trained with a learning rate of 1e-4 and batch size 8 for 5 epochs using the Adam optimiser [51]. k_{prompt} is set to 100 for all domains. The *Response Generator* is trained with a learning rate of 1e-4 and batch size of 8 using Adam optimiser with warmup for 1,000 steps. Our choice of these hyperparameters remains constant across all the domains. k_{know} is set to 7, 6 and 3 for the *film*, *music* and *travel* domains respectively. We utilise beam search decoding with beam width 5 for response generation. It is essential to note that we conduct a two-stage training of PICKD. In the first stage, the *Knowledge Selector* is trained to retrieve appropriate facts. Once this learning phase is complete, the *Response Generator* is trained for knowledgeable response generation.

3 https://huggingface.co/

Table 8.2: Performance of different methodologies on domain-grounded datasets for knowledgeable dialogue generation. We report the performance of PICKD using the average of 5 different runs of the entire framework. The results are statistically significant with p<0.01

Model	PPL	BLEU-1/2/3/4				Dist-1/2/3/4			
Film									
Seq2Seq	23.88	26.97	14.31	8.53	5.30	2.32	6.13	10.88	16.14
HRED	24.74	27.03	14.07	8.30	5.07	2.55	7.35	14.12	21.86
Seq2Seq+Know	25.56	27.45	14.51	8.66	5.32	2.85	7.98	15.09	23.17
HRED+Know	26.27	27.94	14.69	8.73	5.40	2.86	8.08	15.81	24.93
BART	**2.66**	28.54	13.28	14.21	11.00	2.46	14.12	25.72	36.12
RT-KGD	2.86	32.11	22.21	16.68	13.18	**3.05**	16.34	31.36	44.68
PICKD$_{\text{ablated}}$	5.36	38.31	30.66	26.26	22.84	2.79	17.35	31.56	43.61
PICKD	4.89	**40.69**	**33.11**	**28.58**	**25.46**	2.96	**19.08**	**35.23**	**48.97**
Music									
Seq2Seq	16.17	28.89	16.56	10.63	7.17	2.52	7.02	12.69	18.78
HRED	16.82	29.92	17.31	11.17	7.52	2.71	7.71	14.07	20.97
Seq2Seq+Know	17.12	29.60	17.26	11.36	7.84	3.93	12.35	23.01	34.23
HRED+Know	17.69	29.73	17.51	11.59	8.04	3.80	11.7	22.00	33.37
BART	**2.46**	31.65	23.04	18.22	15.05	2.80	13.69	24.73	34.59
RT-KGD	2.47	40.75	31.26	25.56	21.64	**4.18**	**17.38**	30.05	41.05
PICKD$_{\text{ablated}}$	5.94	38.01	30.86	26.57	23.59	3.07	15.12	29.19	40.65
PICKD	4.55	**41.56**	**33.81**	**28.50**	**25.97**	2.84	16.17	**31.10**	**43.77**
Travel									
Seq2Seq	10.44	29.61	20.04	14.91	11.74	3.75	11.15	19.01	27.16
HRED	10.90	30.92	20.97	15.61	12.30	4.15	12.01	20.52	28.74
Seq2Seq+Know	10.62	37.04	27.28	22.16	18.94	4.25	13.64	24.18	34.35
HRED+Know	11.15	36.87	26.68	21.31	17.96	3.98	13.31	24.06	34.35
BART	1.83	34.77	29.11	25.69	23.33	2.70	13.39	21.92	29.53
RT-KGD	**1.61**	47.56	41.46	37.40	34.31	**3.58**	15.50	26.1	35.72
PICKD$_{\text{ablated}}$	3.40	46.68	41.23	37.90	35.53	2.46	16.55	26.88	34.74
PICKD	2.41	**52.40**	**47.37**	**44.19**	**41.88**	2.92	**17.21**	**27.67**	**36.31**

8.4 RESULTS AND DISCUSSION

This section reports the performance of PICKD on the knowledge-grounded dialogue generation task. We also conduct ablation studies by changing different parameters of PICKD that can potentially impact the performance. For completeness, we analyse the performance of different methodologies on human evaluation metrics and conduct error analysis on the generation results.

8.4.1 Automatic Evaluation

Table 8.2 showcases the performance of different baselines against PICKD . PICKD outperforms all the competing methodologies on the targeted datasets in the BLEU metric. The results indicate that the generated responses have a high textual overlap with the reference ground-truth responses as higher Bleu-N scores indicate good and fluent responses. PICKD outperforms other

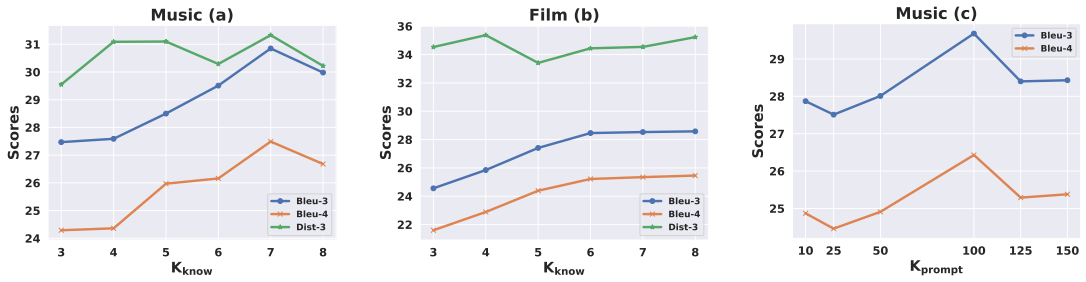

Figure 8.3: Performance impact of PICKD with varying context size (k_{know}) for the *Knowledge Selector* on (a) *Music* and (b) *Film* domain. (c) showcases the performance of PICKD with changing the number of prompt tokens.

methodologies on the Dist-2,3,4 metrics on the *film* and *travel* domains while on the Dist-3,4 metrics on the *music* domain. This indicates the responses generated by PICKD are not bland and are more engaging. Nevertheless, it is interesting to notice that RT-KGD has better perplexity than PICKD over all three domains since during the training of the *Response Generator*, generation is more constrained by the necessity to make factually correct answers over fluent, inaccurate responses. Furthermore, we evaluate the performance of PICKD when the prompt tokens and recency embeddings are removed. During the training of the *Knowledge Selector*, only the classification heads are fine-tuned, resulting in a variant of the model referred to as $\text{PICKD}_{\text{ablated}}$. Our results indicate that the performance of $\text{PICKD}_{\text{ablated}}$ is worse than the original model, highlighting the importance of the prompt tokens and recency embeddings in the PICKD framework.

Overall, we find that PICKD generated responses that have a higher overlap with the ground-truth responses, in turn suggesting the responses are readable and understandable by humans.

8.4.2 Impact of Prompt Length

The *Knowledge Selector* uses our novel *In-Situ* prompt tuning paradigm for retrieving context-relevant knowledge. The module injects prompt tokens within the context-knowledge text, and the prompt tokens along with the classification heads are then trained for the ranking task. A lower number of prompt tokens leads to fewer trainable parameters, thereby limiting the performance of PICKD. In comparison, a higher number of prompt tokens would lead to the truncation of knowledge text due to the limited length processing capacity of the RoBERTa model. Hence, empirically setting the value of K_{prompt} is essential for good performance.

As seen in Figure 8.3c, the performance of PICKD follows a bell-shaped curve with increasing K_{prompt}, illustrating the validity of our hypothesis (as discussed above). Hence, it is essential to choose the value of K_{prompt} which maximises the performance of PICKD in different domains.

8.4.3 Impact of Knowledge Length

The performance of the *Response Generator* relies on the knowledge triples retrieved by the *Knowledge Selector* module. The number of knowledge triples retrieved thus impacts the overall performance of PICKD. Due to this, we conduct an ablation study on the performance of PICKD by varying the knowledge memory size(k_{know}) retrieved by the *Knowledge Selector*.

Table 8.3: Human evaluation results.

Model	Fluency	Coherence
Film/κ	0.78	0.73
HRED+Know	1.74	0.22
RT-KGD	1.89	0.35
PICKD	**1.94**	**1.25**
Music/κ	0.91	0.93
HRED+Know	1.49	0.38
RT-KGD	1.84	0.46
PICKD	**1.91**	**1.15**
Travel/κ	0.95	0.94
HRED+Know	1.35	0.44
RT-KGD	1.64	1.06
PICKD	**1.70**	**1.44**

We fluctuate the value of k_{know} on the film (Figure 8.3a) and *music* (Figure 8.3b) domain. It is to be noticed that in the *music* domains, the performance increases with increasing k_{know} and then decreases with increasing it further. This increase can be explained due to an increase in the number of relevant facts retrieved by the *Knowledge Selector* module. On further increasing K_{know}, the *Response Generator* induces relevant knowledge but is different from the ground truth fact leading to a decline in performance. It is interesting to notice that in the film domain, the model performance stabilises on higher values of k_{know}. This is due to the limited length processing capacity of BART. As the *Response Generator* component concatenates the dialogue context and the knowledge facts together, knowledge gets truncated while processing the input for BART input, resulting in performance saturation.

8.4.4 Manual Evaluation

We conduct evaluations on 50 examples from each of the three domain-specific datasets using two in-house human annotators who are native Chinese speakers. Following Zhou et al. [172], the annotators are asked to rate the generated responses on fluency and coherence using a three-point scale (0, 1, 2). The results, shown in Table 8.3, demonstrate that PICKD outperforms other methods on all datasets, particularly in terms of coherence, indicating its ability to generate responses with relevant knowledge and appropriate context. The high Fleiss' Kappa (κ) [31] indicates the robustness of evaluation due to high inter-annotator agreement showcasing coherence between the annotators.

8.4.5 Error Analysis

This section sheds light on interesting scenarios underlying the internal working of PICKD. Table 5.6 showcases two examples wherein the results produced by PICKD and other baseline models differ from the ground-truth results.

Table 8.4: Response generation analysis of PICKD and other baselines.

Conversation 1	Conversation 2
U1: Have you heard the song "Everyone Will"? U2: I've heard it, it's a very good R&B style song. U1: How long is it?	... U1: Yes, and we know her because she played the second female lead in Blue Bridge of Souls. U2: Oh, and this movie is a classic, and it was released in America. U1: Do you know what year it was released?
True Knowledge: Head="Everyone Will" Relation="Song Duration", Tail="2:58"	True Knowledge: Head="Blue Bridge of Souls" Relation="release time", Tail="May 17, 1940"
Human: 2 minutes and 58 seconds. HRED+Know: The duration is 4 minutes and 46 seconds. RT-KGD: 3 minutes and 14 seconds. PICKD: 2:58, who sang this song?	Human: Yes, 1940. HRED+Know: Yes, it was on September 14, 2011. RT-KGD: You know, it was released on March 25, 1940. PICKD: Yes, it was September 9, 1939.

In the first example, the response generated by PICKD utilises the knowledge triple effectively. The model generates a response that is consistent with the knowledge triple and is grammatically correct. Although such responses are acceptable, they impact the automatic evaluation scores adversely, leading to an underestimation of the model performance. On the other hand, responses generated using other baseline methodologies do not utilise the knowledge effectively and generate factually incorrect responses.

The second example showcases an example wherein the responses generated by the two baselines and PICKD are fluent but factually inconsistent. This is due to the *Knowledge Retriever* selecting incorrect knowledge from the sources. Thus, the *Response Generator* cannot generate the appropriate facts in the response. A similar trend is observed in other baselines. Since our approach incorporates both unstructured and structured knowledge, and the dataset we chose lacks exact annotation of structured knowledge in the responses, the performance analysis is limited for the knowledge selector. It is interesting to notice that although RT-KGD produces factually incorrect responses, it would score better on automatic metrics due to lexical overlap with the human response. Such cases of responses can lead to overestimating performance on automatic metrics.

8.5 SUMMARY

In this chapter, we propose PICKD, a two-stage framework for knowledge-grounded dialogue. PICKD employs the novel *In-Situ* prompt tuning mechanism enabling the selection of appropriate knowledge suited to the dialogue context. The second stage of PICKD engages the BART model as the encoder-decoder framework for knowledgeable response generation. The *knowledge selector* in PICKD selects appropriate knowledge from both structured and unstructured sources of knowledge and *response generator* utilises this retrieved knowledge and the context for generating an appropriate response. We conduct extensive analysis and depict the performance on three domain-specific knowledge-grounded dialogue datasets and exhibit the improved performance in knowledgeable response generation. The PICKD framework facilitates low-parameter fine-tuning of the knowledge selector component in chatbots, while ensuring the generation of informative and coherent responses.

9 CONCLUSIONS

This thesis proposes solutions for addressing various challenges that arise in the development of knowledge-grounded chatbots. Specifically, in Chapter 4, we address the selection of relevant subgraphs from KGs for context-aware integration leading towards improved performance of the *structured knowledge selector* in chatbots. Thereafter, in Chapter 6 we propose self-supervised training of QA components for incorporating unstructured knowledge in chatbots. This enables appropriate knowledge selection from unstructured sources by generating synthetic datasets without the need for manually annotated training datasets. We also suggest a framework for the generation of informative and coherent responses leveraging both structured and unstructured knowledge sources in Chapter 8. Additionally, in Chapter 5, we explore the use of semantically meaningful walk paths over KGs for enhancing conversation explainability.

9.1 DISCUSSION AND CONTRIBUTIONS

The first contribution of this thesis proposed a novel approach to enhance the structured knowledge retrieval performance of the *knowledge selector* (Figure 9.1) component in CRSs. The framework emphasised the importance of constructing task-relevant subgraphs from KGs to propagate information into *knowledge selector*. To this end, we introduced various subgraphs for the task and demonstrate the effectiveness of our model in outperforming the current state-of-the-art on multiple metrics. Our approach leverages pre-trained KG embeddings and positional embeddings. Our experiments demonstrated that our proposed model has superior performance in recommending disconnected entities in the KG and exhibits better performance as the conversation context increases. This contribution is highly beneficial in an industrial scenario wherein users can be recommended relevant products, thereby improving customer satisfaction and potentially an increase in sales.

Thereafter we explored the utilisation of unstructured knowledge in chatbot and its potential for answering user queries. Unstructured knowledge sources are commonly used in a chatbot's QA components to retrieve relevant text spans or sentences that can potentially answer the user query. We propose frameworks for self-supervised training of the EQA and AS2 modules in a chatbot, which can enhance the performance of the *knowledge selector* on unstructured knowledge retrieval tasks (as depicted in Figure 9.1). We introduced self-learning based QA frameworks tailored for closed-domain applications, tackling the challenge of adapting QA modules in chatbots to domains where annotated domain-specific data is scarce and costly. To overcome this challenge, we introduce synthetic data generation using large language models, which facilitates the fine-tuning of pre-trained QA models for improved domain-specific QA platforms. In addition, we propose a novel context retrieval methodology based on semantically relevant contexts via sentence embeddings. We propose *QASAR* that enables the adaptation of EQA modules and

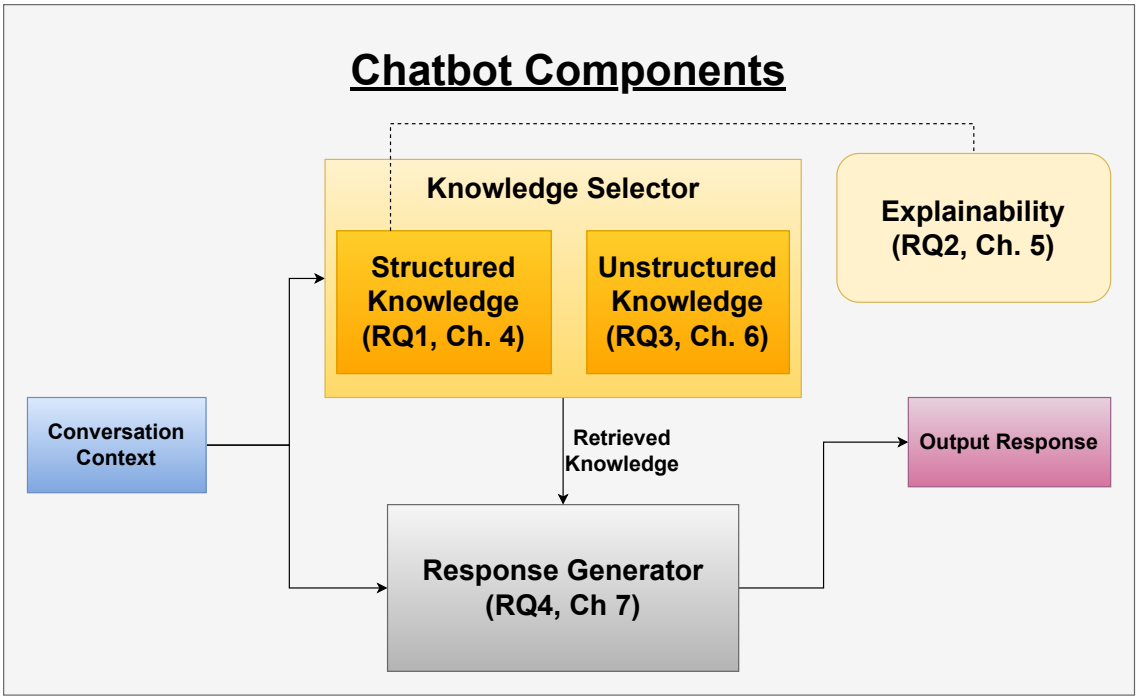

Figure 9.1: Schema of the research questions addressed in this thesis pertaining to knowledge-grounded chatbots.

SEDAN for AS2 using synthetically generated datasets. Extensive experiments on various open- and closed-domain QA datasets demonstrate the effectiveness of our approach, which achieves statistically significant performance improvements compared to existing techniques. The proposed self-supervised frameworks are helpful in an industrial setting as they address the challenges of adapting QA components of chatbots to domain-specific applications where there is a lack of availability and high cost associated with annotating closed-domain data. The utilisation of synthetic datasets in *QASAR* and *SEDAN* for domain-specific QA applications making them cost-effective and scalable. Researchers can further explore the utilisation of synthetically generated datasets from large language models on various other NLP tasks. Furthermore, our framework is shown to be robust and scalable in diverse domain and empirical settings, making it suitable for deployment in enterprise settings.

Subsequently, we present KG-CRuSE, a lightweight LSTM-based framework for explainable conversational reasoning. The task of selecting relevant knowledge to generate informative responses can be framed as path traversal over a knowledge graph conditioned on the dialogue context, with these paths providing an explanation of the conversation's flow. To capture the semantic information in both the dialogue history and knowledge graph elements, we employ sentence embeddings. Through extensive evaluation, we demonstrate that KG-CRUSE outperforms several baseline models in terms of both explainability and response entity retrieval. Researchers in the field of knowledge-grounded dialogue generation can employ the KG-CRuSE framework to gain insights into model behaviour and understand the rationale behind incorporating specific knowledge into the generated responses. This understanding can facilitate improvements in the quality and relevance of knowledge integration in the response text. This framework can be utilised in customer service applications to provide users with relevant explanations for the responses generated for their queries. It can also be utilised in healthcare applications wherein

doctors or patients can converse with a chatbot that can provide explainable answers related to diseases.

Finally, to address the challenges associated with integrating structured and unstructured for generating knowledgeable and informative responses, we introduce *PICKD*, a two-stage framework for knowledge-grounded dialogue, which leverages novel *In-Situ* prompt tuning mechanism to train the *knowledge selector* for selecting appropriate knowledge conditioned on the dialogue context with minimal parameter learning. The *response generator* of *PICKD* employs the BART model as an encoder-decoder framework for knowledgeable response generation conditioned on the context and the knowledge retrieved by *knowledge selector*. We evaluate the framework on three domain-specific knowledge-grounded dialogue datasets and demonstrate improved performance in generating knowledgeable responses. *PICKD* can be leveraged in customer support scenarios wherein informative and knowledgeable responses are generated using external knowledge sources pertinent to the dialogue context. This can be particularly beneficial in various applications, such as movie recommendation, travel-related queries, or music recommendations, which can lead to efficient and effective customer service, resulting in enhanced customer satisfaction and reduced human workload.

This thesis tackles various obstacles related to distinct modules of a knowledge-grounded chatbot and proposes robust solutions for constructing effective knowledge-grounded chatbots.

9.2 FUTURE DIRECTIONS

While this thesis offers solutions to several challenges encountered in the development of knowledge-grounded chatbots, it also highlights a number of potential research directions to further advance the state-of-the-art.

- Chapter 4 demonstrated the importance of contextually relevant subgraphs for enhancing the recommendation performance of a CRS system. However, the investigation into the influence of contextually relevant subgraphs on the response generation component of the CRS has not been explored. Can we create contextually relevant subgraphs by leveraging the semantic information in the dialogue history to enhance the quality of the response generator in the CRS system?

- The use of synthetic datasets has enabled training AS2 and EQA systems without requiring manually annotated training data. However, the quality of these synthetically generated datasets relies on the LMs used for data generation. Fine-tuning larger language models for data generation can be costly and time-consuming. To address this, can we leverage discrete prompts as input to large language models to generate high-quality synthetic datasets?

- Training chatbots with relevant information from KGs requires datasets that have KG-grounded conversations. However, developing open/closed domain KG-conversations parallel datasets can be challenging, time-consuming and expensive. As walk-paths over a KG are indicative of semantic shifts in topics, can we generate synthetic conversations from such paths leveraging large language models for training chatbots without the need for manually annotated synthetic datasets?

- Prompt tuning is a technique that enables the adaptation of a pre-trained frozen language model (LM) for a specific task by tuning learnable prompt tokens. In this context, can we use prompt tuning for end-to-end training of the knowledge selector and response generator for knowledge-grounded response generation?
- Reinforcement Learning is a tool that is used to train agents to learn a specific task by interacting with an environment. It has shown to be useful to construct walk-paths over a KG to tackle NLP tasks. This begs the question, can we utilise reinforcement learning to construct walk-paths over a KG that can potentially provide an explanation to chatbot conversations?

Addressing these research questions will enable efficient adaptation of chatbots for domain-specific applications with limited training data and compute requirements.

BIBLIOGRAPHY

[1] Sameera A Abdul-Kader and John C Woods. "Survey on chatbot design techniques in speech conversation systems". In: *International Journal of Advanced Computer Science and Applications* 6.7 (2015).

[2] Eleni Adamopoulou and Lefteris Moussiades. "An Overview of Chatbot Technology". In: *Artificial Intelligence Applications and Innovations - 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5-7, 2020, Proceedings, Part II*. Ed. by Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis. Vol. 584. IFIP Advances in Information and Communication Technology. Springer, 2020, pp. 373–383.

[3] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. "Towards a Human-like Open-Domain Chatbot". In: *CoRR* abs/2001.09977 (2020). arXiv: 2001.09977. URL: https://arxiv.org/abs/2001.09977.

[4] Abien Fred Agarap. "Deep Learning using Rectified Linear Units (ReLU)". In: *CoRR* abs/1803.08375 (2018). arXiv: 1803.08375.

[5] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. "Synthetic QA Corpora Generation with Roundtrip Consistency". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 6168–6173.

[6] Giambattista Amati. "BM25". In: *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009, pp. 257–260. ISBN: 978-0-387-39940-9.

[7] Haytham Assem, Sourav Dutta, and Edward Burgin. "DTAFA: Decoupled Training Architecture for Efficient FAQ Retrieval". In: *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2021, Singapore and Online, July 29-31, 2021*. Ed. by Haizhou Li, Gina-Anne Levow, Zhou Yu, Chitralekha Gupta, Berrak Sisman, Siqi Cai, David Vandyke, Nina Dethlefs, Yan Wu, and Junyi Jessy Li. Association for Computational Linguistics, 2021, pp. 423–430.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Ed. by Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux. Vol. 4825. Lecture Notes in Computer Science. Springer, 2007, pp. 722–735.

[9] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. "Fast Incremental and Personalized PageRank". In: *Proc. VLDB Endow.* 4.3 (2010), pp. 173–184.

[10] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. "TuckER: Tensor Factorization for Knowledge Graph Completion". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 5184–5193.

[11] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. "The Pushshift Reddit Dataset". In: *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*. Ed. by Munmun De Choudhury, Rumi Chunara, Aron Culotta, and Brooke Foucault Welles. AAAI Press, 2020, pp. 830–839. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/7347.

[12] George Bebis and Michael Georgiopoulos. "Feed-forward neural networks". In: *IEEE Potentials* 13.4 (1994), pp. 27–31.

[13] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. Ed. by Jason Tsong-Li Wang. ACM, 2008, pp. 1247–1250.

[14] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger. 2013, pp. 2787–2795.

[15] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. Ed. by Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton. The Association for Computational Linguistics, 2015, pp. 632–642.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020.

[17] Ying-Hong Chan and Yao-Chung Fan. "A Recurrent BERT-based Model for Question Generation". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*. Ed. by Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Association for Computational Linguistics, 2019, pp. 154–162.

[18] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. "Towards Knowledge-Based Recommender Dialog System". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019, pp. 1803–1813.

[19] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. "Improving efficiency and accuracy in multilingual entity extraction". In: *I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, I-SEMANTICS '13, Graz, Austria, September 4-6, 2013*. Ed. by Marta Sabou, Eva Blomqvist, Tommaso Di Noia, Harald Sack, and Tassilo Pellegrini. ACM, 2013, pp. 121–124.

[20] Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Wai Lam, and Ying Shen. "Contextualized Knowledge-Aware Attentive Neural Network: Enhancing Answer Selection with Knowledge". In: *ACM Trans. Inf. Syst.* 40.1 (Sept. 2021). ISSN: 1046-8188.

[21] Yang Deng, Wenxuan Zhang, and Wai Lam. "Learning to Rank Question Answer Pairs with Bilateral Contrastive Data Augmentation". In: *Proceedings of the Seventh Workshop on Noisy User-generated Text, W-NUT 2021, Online, November 11, 2021*. Ed. by Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi. Association for Computational Linguistics, 2021, pp. 175–181.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186.

[23] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. "Wizard of Wikipedia: Knowledge-Powered Conversational Agents". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[24] David Dominguez-Sal and Mihai Surdeanu. "A Machine Learning Approach for Factoid Question Answering". In: *Proces. del Leng. Natural* 37 (2006).

[25] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. "DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 2368–2378.

[26] Ondrej Dusek and Filip Jurcícek. "A Context-aware Natural Language Generator for Dialogue Systems". In: *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*. The Association for Computer Linguistics, 2016, pp. 185–190.

[27] Jeffrey L. Elman. "Finding Structure in Time". In: *Cogn. Sci.* 14.2 (1990), pp. 179–211.

[28] "Exploring Neural Net Augmentation to BERT for Question Answering on SQUAD 2.0". 2019. arXiv: 1908.01767.

[29] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. "Building Watson: An Overview of the DeepQA Project". In: *AI Mag.* 31.3 (2010), pp. 59–79.

[30] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. "MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*. Ed. by Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Association for Computational Linguistics, 2019, pp. 1–13.

[31] Joseph L Fleiss. "Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5 (1971), p. 378.

[32] Tingchen Fu, Xueliang Zhao, Chongyang Tao, Ji-Rong Wen, and Rui Yan. "There Are a Thousand Hamlets in a Thousand People's Eyes: Enhancing Knowledge-grounded Dialogue with Personal Memory". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, 2022, pp. 3901–3913.

[33] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, and Gerard de Melo. "Fairness-Aware Explainable Recommendation over Knowledge Graphs". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, 2020, pp. 69–78.

[34] Amati G. "BM25". In: *Encyclopedia of Database Systems*. Springer, 2009.

[35] Matteo Gabburo, Rik Koncel-Kedziorski, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. "Knowledge Transfer from Answer Ranking to Answer Generation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, 2022, pp. 9481–9495.

[36] Luyu Gao, Zhuyun Dai, and Jamie Callan. "Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline". In: *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II*. Ed. by Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast,

and Fabrizio Sebastiani. Vol. 12657. Lecture Notes in Computer Science. Springer, 2021, pp. 280–286.

[37] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. "TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 7780–7788.

[38] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. "A Knowledge-Grounded Neural Conversation Model". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 5110–5117.

[39] Rujun Han, Luca Soldaini, and Alessandro Moschitti. "Modeling Context in Answer Sentence Selection Systems on a Latency Budget". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Association for Computational Linguistics, 2021, pp. 3005–3010.

[40] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), pp. 1735–1780.

[41] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers, 2021.

[42] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. "Knowledge Graphs". In: *ACM Comput. Surv.* 54.4 (2022), 71:1–71:37.

[43] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020.

[44] Weiyi Huang, Jiahao Jiang, Qiang Qu, and Min Yang. "AILA: A Question Answering System in the Legal Domain". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. Ed. by Christian Bessiere. ijcai.org, 2020, pp. 5258–5260.

[45] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Trans. Assoc. Comput. Linguistics* 8 (2020), pp. 64–77.

[46] Jaehun Jung, Bokyung Son, and Sungwon Lyu. "AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, 2020, pp. 3484–3497.

[47] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement Learning: A Survey". In: *J. Artif. Intell. Res.* 4 (1996), pp. 237–285.

[48] Sekitoshi Kanai, Yasuhiro Fujiwara, and Sotetsu Iwamura. "Preventing Gradient Explosions in Gated Recurrent Units". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 435–444.

[49] Aniruddha Kembhavi, Min Joon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. "Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5376–5384.

[50] Taejin Kim, Yeoil Yun, and Namgyu Kim. "Deep learning-based knowledge graph generation for COVID-19". In: *Sustainability* 13.4 (2021), p. 2276.

[51] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[52] Alina Kramchaninova and Arne Defauw. "Synthetic Data Generation for Multilingual Domain-Adaptable Question Answering Systems". In: *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, EAMT 2022, Ghent, Belgium, June 1-3, 2022*. Ed. by Helena Moniz, Lieve Macken, Andrew Rufener, Loïc Barrault, Marta R. Costa-jussà, Christophe Declercq, Maarit Koponen, Ellie Kemp, Spyridon Pilos, Mikel L. Forcada, Carolina Scarton, Joachim Van den Bogaert, Joke Daems, Arda Tezcan, Bram Vanroy, and Margot Fonteyne. European Association for Machine Translation, 2022, pp. 151–160.

[53] Nikolaus Kriegeskorte and Tal Golan. "Neural network models and deep learning". In: *Current Biology* 29.7 (2019), R231–R236.

[54] Jonás Kulhánek, Vojtech Hudecek, Tomás Nekvinda, and Ondrej Dusek. "AuGPT: Dialogue with Pre-trained Language Models and Data Augmentation". In: *CoRR* abs/2102.05126 (2021). arXiv: 2102.05126.

[55] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. "Natural Questions: a Benchmark for Question Answering Research". In: *Trans. Assoc. Comput. Linguistics* 7 (2019), pp. 452–466.

[56] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[57] Md. Tahmid Rahman Laskar, Jimmy Xiangji Huang, and Enamul Hoque. "Contextualized Embeddings based Transformer Encoder for Sentence Similarity Modeling in Answer Selection Task". In: *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, 2020, pp. 5505–5514.

[58] Ivano Lauriola and Alessandro Moschitti. "Answer Sentence Selection Using Local and Global Context in Transformer Models". In: *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*. Ed. by Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani. Vol. 12656. Lecture Notes in Computer Science. Springer, 2021, pp. 298–312.

[59] Edward E Leamer. *Specification searches: Ad hoc inference with nonexperimental data*. Vol. 53. John Wiley & Sons Incorporated, 1978.

[60] Hyeon-gu Lee, Youngjin Jang, and Harksoo Kim. "Machine Reading Comprehension Framework Based on Self-Training for Domain Adaptation". In: *IEEE Access* 9 (2021), pp. 21279–21285.

[61] Brian Lester, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 3045–3059.

[62] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, 2020, pp. 7871–7880.

[63] Hongyu Li, Xiyuan Zhang, Yibing Liu, Yiming Zhang, Quan Wang, Xiangyang Zhou, Jing Liu, Hua Wu, and Haifeng Wang. "D-NET: A Pre-Training and Fine-Tuning Framework for Improving the Generalization of Machine Reading Comprehension". In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*. Ed. by Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Association for Computational Linguistics, 2019, pp. 212–219.

[64] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. "Towards Deep Conversational Recommendations". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Process-*

ing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 9748–9758.

[65] Yang Li and Tao Yang. "Word embedding for understanding natural language: a survey". In: *Guide to big data applications* (2018), pp. 83–104.

[66] Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. "Pre-training Transformer Models with Sentence-Level Objectives for Answer Sentence Selection". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, 2022, pp. 11806–11816.

[67] Dongding Lin, Jian Wang, and Wenjie Li. "COLA: Improving Conversational Recommender Systems by Collaborative Augmentation". In: *CoRR* abs/2212.07767 (2022). arXiv: 2212.07767.

[68] Xi Victoria Lin, Richard Socher, and Caiming Xiong. "Multi-Hop Knowledge Graph Reasoning with Reward Shaping". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 3243–3253.

[69] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. "Learning Entity and Relation Embeddings for Knowledge Graph Completion". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. Ed. by Blai Bonet and Sven Koenig. AAAI Press, 2015, pp. 2181–2187. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571.

[70] Qi Liu, Matt J. Kusner, and Phil Blunsom. "A Survey on Contextual Embeddings". In: *CoRR* abs/2003.07278 (2020). arXiv: 2003.07278. URL: https://arxiv.org/abs/2003.07278.

[71] Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. "Neural Machine Reading Comprehension: Methods and Trends". In: *CoRR* abs/1907.01118 (2019). arXiv: 1907.01118.

[72] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692.

[73] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. "Knowledge Aware Conversation Generation with Explainable Reasoning over Augmented Graphs". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 1782–1792.

[74] Zihan Liu, Mostofa Patwary, Ryan Prenger, Shrimai Prabhumoye, Wei Ping, Mohammad Shoeybi, and Bryan Catanzaro. "Multi-Stage Prompting for Knowledgeable Dialogue Generation". In: *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, 2022, pp. 1317–1337.

[75] Ekaterina Loginova, Stalin Varanasi, and Günter Neumann. "Towards Multilingual Neural Question Answering". In: *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshops, AI*QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2-5, 2018, Proceedings*. Ed. by András Benczúr, Bernhard Thalheim, Tomás Horváth, Silvia Chiusano, Tania Cerquitelli, Csaba István Sidló, and Peter Z. Revesz. Vol. 909. Communications in Computer and Information Science. Springer, 2018, pp. 274–285.

[76] Jing Lu, Gustavo Hernández Ábrego, Ji Ma, Jianmo Ni, and Yinfei Yang. "Multi-stage Training with Improved Negative Contrast for Neural Passage Retrieval". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 6091–6103.

[77] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. "RevCore: Review-Augmented Conversational Recommendation". In: *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Vol. ACL/IJCNLP 2021. Findings of ACL. Association for Computational Linguistics, 2021, pp. 1161–1173.

[78] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. "Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 3219–3232.

[79] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. "On Measuring Social Biases in Sentence Encoders". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 622–628.

[80] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2013.

[81] Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997. ISBN: 978-0-07-042807-2. URL: https://www.worldcat.org/oclc/61321007.

[82] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. "Towards Exploiting Background Knowledge for Building Conversation Systems". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 2322–2332.

[83] Diego Mollá and José Luis Vicedo González. "Question Answering in Restricted Domains: An Overview". In: *Comput. Linguistics* 33.1 (2007), pp. 41–61.

[84] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. "OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 845–854.

[85] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. Ed. by Johannes Fürnkranz and Thorsten Joachims. Omnipress, 2010, pp. 807–814.

[86] Sridhar Narayan. "The Generalized Sigmoid Activation Function: Competetive Supervised Learning". In: *Inf. Sci.* 99.1-2 (1997), pp. 69–82.

[87] Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Paliouras. "Results of the Seventh Edition of the BioASQ Challenge". In: *Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD*. Ed. by Peggy Cellier and Kurt Driessens. Vol. 1168. Communications in Computer and Information Science. Springer, 2019, pp. 553–568.

[88] Jinjie Ni, Vlad Pandelea, Tom Young, Haicang Zhou, and Erik Cambria. "HiTKG: Towards Goal-Oriented Conversations via Multi-Hierarchy Learning". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 11112–11120.

[89] Mohammad Nuruzzaman and Omar Khadeer Hussain. "A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks". In: *15th IEEE International Conference on e-Business Engineering, ICEBE 2018, Xi'an, China, October 12-14, 2018*. IEEE Computer Society, 2018, pp. 54–61.

[90] Ibrahim Burak Özyurt. "End-to-end Biomedical Question Answering via Bio-AnswerFinder and Discriminative Language Representation Models". In: *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*. Ed. by Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, and Florina Piroi. Vol. 2936. CEUR Workshop Proceedings. CEUR-WS.org, 2021, pp. 286–301.

[91] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[92] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 2002, pp. 311–318.

[93] Dimitris Pappas, Prodromos Malakasiotis, and Ion Androutsopoulos. "Data Augmentation for Biomedical Factoid Question Answering". In: *Proceedings of the 21st Workshop on Biomedical Language Processing, BioNLP@ACL 2022, Dublin, Ireland, May 26, 2022*. Ed. by Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii. Association for Computational Linguistics, 2022, pp. 63–81.

[94] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. "A Decomposable Attention Model for Natural Language Inference". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, 2016, pp. 2249–2255.

[95] Prasanna Parthasarathi and Joelle Pineau. "Extending Neural Generative Conversational Model using External Knowledge Sources". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 690–695.

[96] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. "TF-Ranking: Scalable TensorFlow Library for Learning-to-Rank". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*. ACM, 2019, pp. 2970–2978.

[97] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, 2014, pp. 1532–1543.

[98] Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. "Training Question Answering Models From Synthetic Data". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, 2020, pp. 5811–5826.

[99] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018.

[100] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[101] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[102] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, 2016, pp. 2383–2392.

[103] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 3980–3990.

[104] Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. "Synthetic Data Augmentation for Zero-Shot Cross-Lingual Question Answering". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 7016–7030.

[105] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. "Recipes for Building an Open-Domain Chatbot". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Association for Computational Linguistics, 2021, pp. 300–325.

[106] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *CoRR* abs/1609.04747 (2016). arXiv: 1609.04747. URL: http://arxiv.org/abs/1609.04747.

[107] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR* abs/1910.01108 (2019).

[108] Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. "Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, 2020, pp. 4498–4507.

[109] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. Ed. by Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam. Vol. 10843. Lecture Notes in Computer Science. Springer, 2018, pp. 593–607.

[110] Sina J Semnani, Madhulima Pandey, and Manish Pandey. "Domain-Specific Question Answering at Scale for Conversational Systems". In: *3rd NeurIPS Conversational AI Workshop, 33rd Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 1–10.

[111] Yeon Seonwoo, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. *Context-Aware Answer Extraction in Question Answering*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. 2020.

[112] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. Ed. by Dale Schuurmans and Michael P. Wellman. AAAI Press, 2016, pp. 3776–3784.

[113] Aliaksei Severyn and Alessandro Moschitti. "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*. Ed. by Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto. ACM, 2015, pp. 373–382.

[114] Siamak Shakeri, Noah Constant, Mihir Kale, and Linting Xue. "Towards Zero-Shot Multilingual Synthetic Question and Answer Generation for Cross-Lingual Reading Comprehension". In: *Proceedings of the 14th International Conference on Natural Language Generation, INLG 2021, Aberdeen, Scotland, UK, 20-24 September, 2021*. Ed. by Anya Belz, Angela Fan, Ehud Reiter, and Yaji Sripada. Association for Computational Linguistics, 2021, pp. 35–45.

[115] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. "Fast WordPiece Tokenization". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 2089–2103.

[116] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. "A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion". In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*. Ed. by James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu. ACM, 2015, pp. 553–562.

[117] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. "A Neural Network Approach to Context-Sensitive Generation of Conversational Responses". In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. Ed. by Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar. The Association for Computational Linguistics, 2015, pp. 196–205.

[118] Robyn Speer, Joshua Chin, and Catherine Havasi. "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, 2017, pp. 4444–4451.

[119] Student. "The probable error of a mean". In: *Biometrika* (1908), pp. 1–25.

[120] Weichen Sun, Fei Su, and Leiquan Wang. "Improving deep neural networks with multi-layer maxout networks and a novel initialization method". In: *Neurocomputing* 278 (2018), pp. 34–40.

[121] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger. 2014, pp. 3104–3112.

[122] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. "MultiModalQA: complex question answering over text, tables and images". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[123] Chuanqi Tan, Furu Wei, Qingyu Zhou, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. "Context-Aware Answer Sentence Selection With Hierarchical Gated Recurrent Neural Networks". In: *IEEE ACM Trans. Audio Speech Lang. Process.* 26.3 (2018), pp. 540–549.

[124] Ambuj Tewari and Peter L. Bartlett. "On the Consistency of Multiclass Classification Methods". In: *J. Mach. Learn. Res.* 8 (2007), pp. 1007–1025.

[125] Ilaria Tiddi and Stefan Schlobach. "Knowledge graphs as tools for explainable machine learning: A survey". In: *Artif. Intell.* 302 (2022), p. 103627.

[126] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*. Ed. by Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay B. Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih. Association for Computational Linguistics, 2017, pp. 191–200.

[127] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*. Ed. by Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay B. Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih. Association for Computational Linguistics, 2017, pp. 191–200.

[128] Yi-Lin Tuan, Sajjad Beygi, Maryam Fazel-Zarandi, Qiaozi Gao, Alessandra Cervone, and William Yang Wang. "Towards Large-Scale Interpretable Knowledge Graph Reasoning for Dialogue Systems". In: *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, 2022, pp. 383–395.

[129] Ledyard R Tucker. "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3 (1966), pp. 279–311.

[130] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 5998–6008.

[131] Nikhita Vedula and Srinivasan Parthasarathy. "FACE-KEG: Fact Checking Explained using KnowledgE Graphs". In: *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*. Ed. by Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich. ACM, 2021, pp. 526–534.

[132] Mani Vegupatti, Matthias Nickles, and Bharathi Raja Chakravarthi. "Simple Question Answering Over a Domain-Specific Knowledge Graph using BERT by Transfer Learning". In: *Proceedings of The 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Republic of Ireland, December 7-8, 2020*. Ed. by Luca Longo, Lucas Rizzo, Elizabeth Hunter, and Arjun Pakrashi. Vol. 2771. CEUR Workshop Proceedings. CEUR-WS.org, 2020, pp. 289–300.

[133] Kexin Wang, Zhixu Li, Jiaan Wang, Jianfeng Qu, Ying He, An Liu, and Lei Zhao. "RT-KGD: Relation Transition Aware Knowledge-Grounded Dialogue Generation". In: *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*. Ed. by Ulrike Sattler, Aidan Hogan, C. Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato. Vol. 13489. Lecture Notes in Computer Science. Springer, 2022, pp. 319–335.

[134] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. "Explainable Reasoning over Knowledge Graphs for Recommendation". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 5329–5336.

[135] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. "Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning". In: *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. Ed. by Aidong Zhang and Huzefa Rangwala. ACM, 2022, pp. 1929–1937.

[136] Yu-An Wang and Yun-Nung Chen. "What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, 2020, pp. 6840–6849.

[137] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve J. Young. "A Network-based End-to-End Trainable Task-oriented Dialogue System". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7,*

2017, Volume 1: Long Papers. Ed. by Mirella Lapata, Phil Blunsom, and Alexander Koller. Association for Computational Linguistics, 2017, pp. 438–449.

[138] Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. "Neural Domain Adaptation for Biomedical Question Answering". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*. Ed. by Roger Levy and Lucia Specia. Association for Computational Linguistics, 2017, pp. 281–289.

[139] Adina Williams, Nikita Nangia, and Samuel R. Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, 2018, pp. 1112–1122.

[140] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. "Session-Based Recommendation with Graph Neural Networks". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 346–353.

[141] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. "Proactive Human-Machine Conversation with Explicit Conversation Goal". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 3794–3804.

[142] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144.

[143] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Trans. Neural Networks Learn. Syst.* 32.1 (2021), pp. 4–24.

[144] Wei Xu and Alex Rudnicky. "Can artificial neural networks learn language models?" In: *Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000, Beijing, China, October 16-20, 2000*. ISCA, 2000, pp. 202–205.

[145] Weiwen Xu, Yang Deng, Huihui Zhang, Deng Cai, and Wai Lam. "Exploiting Reasoning Chains for Multi-hop Science Question Answering". In: *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 1143–1156.

[146] Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. "Dynamic Semantic Graph Construction and Reasoning for Explainable Multi-hop Science Question Answering". In: *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Vol. ACL/IJCNLP 2021. Findings of ACL. Association for Computational Linguistics, 2021, pp. 1044–1056.

[147] Rui Yan. ""Chitty-Chitty-Chat Bot": Deep Learning for Conversational AI". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 5520–5526.

[148] Rui Yan and Dongyan Zhao. "Coupled Context Modeling for Deep Chit-Chat: Towards Conversations between Human and Computer". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. Ed. by Yike Guo and Faisal Farooq. ACM, 2018, pp. 2574–2583.

[149] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. "Building Task-Oriented Dialogue Systems for Online Shopping". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, 2017, pp. 4618–4626.

[150] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. "Embedding Entities and Relations for Learning and Inference in Knowledge Bases". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[151] Chenxu Yang, Zheng Lin, Jiangnan Li, Fandong Meng, Weiping Wang, Lanrui Wang, and Jie Zhou. "TAKE: Topic-shift Aware Knowledge sElection for Dialogue Generation". In: *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*. Ed. by Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na. International Committee on Computational Linguistics, 2022, pp. 253–265.

[152] Yi Yang, Wen-tau Yih, and Christopher Meek. "WikiQA: A Challenge Dataset for Open-Domain Question Answering". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. Ed. by Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton. The Association for Computational Linguistics, 2015, pp. 2013–2018.

[153] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 2369–2380.

[154] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering". In: *Proceedings of the 2021 Conference of the North American Chapter of the Associ-*

ation for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Association for Computational Linguistics, 2021, pp. 535–546.

[155] Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. "Propagate-Selector: Detecting Supporting Sentences for Question Answering via Graph Neural Networks". In: *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, 2020, pp. 5400–5407.

[156] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. "Augmenting End-to-End Dialogue Systems With Commonsense Knowledge". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 4970–4977.

[157] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. "Deep Learning for Answer Sentence Selection". In: *NIPS Deep Learning Workshop*. 2014.

[158] Wenhao Yu, Lingfei Wu, Yu Deng, Qingkai Zeng, Ruchi Mahindru, Sinem Güven, and Meng Jiang. "Technical Question Answering across Tasks and Domains". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, NAACL-HLT 2021, Online, June 6-11, 2021*. Ed. by Young-bum Kim, Yunyao Li, and Owen Rambow. Association for Computational Linguistics, 2021, pp. 178–186.

[159] C. Zeng, S. Li, Q. Li, J. Hu, and J. Hu. "A Survey on Machine Reading Comprehension – Tasks, Evaluation Metrics and Benchmark Datasets". In: *Applied Sciences* 10.7640 (2020), pp. 1–57.

[160] Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. "Fixup Initialization: Residual Learning Without Normalization". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[161] Shi-Wei Zhang, Yiyang Du, Guanzhong Liu, Zhao Yan, and Yunbo Cao. "G4: Grounding-guided Goal-oriented Dialogues Generation with Multiple Documents". In: *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, DialDoc@ACL 2022, Dublin, Ireland, May 26, 2022*. Ed. by Song Feng, Hui Wan, Caixia Yuan, and Han Yu. Association for Computational Linguistics, 2022, pp. 108–114.

[162] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: (2020).

[163] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. "DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation". In: *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020. Ed. by Asli Celikyilmaz and Tsung-Hsien Wen. Association for Computational Linguistics, 2020, pp. 270–278.

[164] Z. Zhang, H. Zhao, and R. Wang. "Machine Reading Comprehension: The Role of Contextualized Language Models and Beyond". In: *Computational Linguistics* 1.1 (2020), pp. 1–51.

[165] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. "Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, 2020, pp. 239–248.

[166] Xiangyu Zhao, Longbiao Wang, Ruifang He, Ting Yang, Jinxin Chang, and Ruifang Wang. "Multiple Knowledge Syncretic Transformer for Natural Dialogue Generation". In: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. Ed. by Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen. ACM / IW3C2, 2020, pp. 752–762.

[167] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. "Knowledge-Grounded Dialogue Generation with Pre-trained Language Models". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, 2020, pp. 3377–3390.

[168] Chujie Zheng and Minlie Huang. "Exploring Prompt-based Few-shot Learning for Grounded Dialog Generation". In: *CoRR* abs/2109.06513 (2021).

[169] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. "A Pre-Training Based Personalized Dialogue Generation Model with Persona-Sparse Data". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 9693–9700.

[170] Hao Zhou, Minlie Huang, Yong Liu, Wei Chen, and Xiaoyan Zhu. "EARL: Informative Knowledge-Grounded Conversation Generation with Entity-Agnostic Representation Learning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 2383–2395.

[171] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. "Commonsense Knowledge Aware Conversation Generation with Graph Attention". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 4623–4629.

[172] Hao Zhou, Chujie Zheng, Kaili Huang, Minlie Huang, and Xiaoyan Zhu. "KdConv: A Chinese Multi-domain Dialogue Dataset Towards Multi-turn Knowledge-driven Conversation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, 2020, pp. 7098–7108.

[173] Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. "A Dataset for Document Grounded Conversations". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii. Association for Computational Linguistics, 2018, pp. 708–713.

[174] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. "Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion". In: *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. Ed. by Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash. ACM, 2020, pp. 1006–1014.

[175] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. "Towards Topic-Guided Conversational Recommender System". In: *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*. Ed. by Donia Scott, Núria Bel, and Chengqing Zong. International Committee on Computational Linguistics, 2020, pp. 4128–4139.

[176] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. "Flexible End-to-End Dialogue System for Knowledge Grounded Conversation". In: *CoRR* abs/1709.04264 (2017).

[177] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books". In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 19–27.