ISLAMIC UNIVERSITY OF TECHNOLOGY

Dialog Generation with Conversational Agent in the Context of Task-Oriented using a Transformer Architecture

By

Faysal Mounir Petouo (180041149)
Yaya Issa Arafat (180041155)

Supervisor
Prof. Dr. Md. Kamrul Hasan
Professor
Dept. of CSE, IUT

Co-Supervisor
Dr. Hasan Mahmud
Associate Professor
Dept. of CSE, IUT

Systems and Software Lab (SSL)
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
A Subsidiary Organ of the Organization of Islamic Cooperation.

A thesis submitted in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering

Academic Year: 2021-2022
May 2023

Declaration of Authorship

Us, Faysal Mounir Petouo Nkayouen and Yaya Issa Arafat, declare that this thesis titled, 'Dialog Generation with Conversational Agent in the Context Of Task-Oriented using a Transformer Architecture' and the work presented in it is our own. We confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Any part of this thesis has not been submitted for any other degree or qualification at this University or any other institution.
- Where we have consulted the published work of others, this is always clearly attributed.

Faysal Mounir Petouo Nkayouen
Student ID - **180041149**

Yaya Issa Arafat
Student ID - **180041155**

Supervisors:

Md. Kamrul Hasan

Prof. Dr. Md. Kamrul Hasan
Professor
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Dr. Hasan Mahmud
Associate Professor
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Contents

List of Figures

List of Tables

Abstract

The use of conversational agents has become increasingly popular in recent years due to their ability to mimic human-like interactions in Human Computer Interaction (HCI) and provide personalized assistance to users. However, creating effective dialogues between humans and conversational agents remains a challenging task, particularly in the context of task-oriented applications. This is because such applications require agents to understand complex user requests and generate appropriate responses that take into account the user's goals, preferences, and constraints.To address this challenge, we propose to adapt the LongT5 (Long Text-To-Text-Transfer Transformer) architecture, a transformer-based language processing model well known for its performance in a lot of Natural Language Processing (NLP) tasks. Then, to explore the use of the new proposed model named MegaT for generating task-oriented dialogues between conversational agents and human user. This involves designing and implementing a task-oriented conversational agent trained on annotated dialogues related to specific tasks. The agent's performance will be evaluated using metrics such as belief accuracy, belief loss, response accuracy, and response loss. The results have been analyzed to identify the strengths and weaknesses of the T5 transformer, the current state-of-the-art model in task-oriented dialogue generation . Experimental results demonstrate that MegaT outperforms the T5-based agent in terms of generating accurate, fluent, and coherent responses to user queries, as well as handling longer sequences of text and producing more informative and engaging responses. We also found that our proposed Transient Global attention for task-oriented dialogue systems produce better results than the local attention mechanism used in LongT5 on MultiWoz 2.2 dataset. The thesis aims to contribute to the development of more effective conversational agents by leveraging the LongT5 model for generating high-quality task-oriented dialogues. This Study provides insights into the use of this recent transformer model and paves the way for further advancements in the field of dialogue generation with conversational agents. . Furthermore, it opens new avenues for future research in the field of dialogue generation with conversational agents.

Keywords: natural language processing; conversational agents; task-oriented dialogue; longer sequences; models; transformers; HCI.

Chapter 1

Introduction

With several applications ranging from sentiment analysis and text summarization to machine translation and speech recognition, the topic of natural language processing (NLP) has long been a fascinating one of machine learning (ML). The creation of Task-Oriented Dialog Systems (TOD), which help users complete specific tasks like hotel, airplane, and restaurant reservations, is one of the biggest problems in the field. Due to its potential to enhance user experience and productivity in a variety of applications, task-oriented dialog systems have attracted a lot of interest recently. These systems' objective is to enable intuitive and natural human-machine communication, which calls for advanced algorithms and architectures that can process demanding user requests and produce pertinent responses. There have been two basic proposals for task-oriented dialog system architecture: the pipeline approach and the end-to-end approach. In the pipeline approach, the dialog system is divided into a number of modules that each carry out a single function, such as intent recognition, slot filling, and response production. These modules process the system's input in order, and the ultimate output is created by merging the results of all the modules. On the other hand, the end-to-end approach tries to create a single neural network model that can manage the full dialog process, from input comprehension to answer creation. By doing away with explicit module design, this method enables more effective and efficient interaction between the user and the system.Depending on the particular requirements and constraints, both the pipeline and end-to-end techniques have advantages and disadvantages and are viable for certain applications. The pipeline approach, for instance, may be more suited for systems that demand great precision and flexibility in processing various input and output types. The end-to-end approach, on the other hand, might be more appropriate for systems that are more akin to human communication and are tolerant of some errors and ambiguity.

1.1 Approaches In Designing Task-Oriented Dialog Systems

1.1.1 The pipeline approach

This strategy is a popular architectural design for creating task-oriented dialog systems. The pipeline technique makes it easier to integrate various components and allows for modular development. It allows the system to operate in a structured and orderly manner as it processes user inputs, updates the dialogue state, makes decisions, and generates responses. The dialogue system can be maintained, debugged, and improved more easily if each stage is built and optimized separately. It makes use of a number of modules, each of which is created to carry out a particular function during the dialog generating process. These modules comprise Dialog Policy (DP), Natural Language Generation (NLG), Natural Language Understanding (NLU), and Dialogue State Tracker (DST).

- Natural Language Understanding (NLU) The first module is this one. It transforms user input in the form of natural language into a semantic frame that the machine can understand. The NLU module extracts the pertinent data from user input and represents it in a structured way that the system can readily handle using cutting-edge algorithms and techniques like named entity recognition, part-of-speech tagging, and dependency parsing.

- Dialogue State Tracker (DST) This second module is in charge of monitoring the dialogue's progress. It produces a representation of the dialogue's current state after receiving as input the semantic frames produced by the NLU module. The user's intent, the work being completed at the time, and any pertinent conversational context or history are all included in this state representation.

- Dialog Policy (DP) This third module outputs dialog acts using the state representation created by the DST module. The system can engage with the user by using a set of actions or answers known as ”dialog acts.” The dialog policy determines the right dialog act depending on the current state of the interaction using a variety of techniques, including rule-based systems, machine learning (ML), and reinforcement learning.
- Natural Language Generation (NLG) The last module is in charge of translating the semantic frame produced by the dialog policy into user-friendly natural language. The NLG module creates fluent and logical responses that are appropriate for the dialogue's present state by utilizing cutting-edge approaches like text creation and sentence planning.

Due of its adaptability and modularity, the pipeline technique has been frequently employed in the creation of task-oriented dialog systems. For improved performance and simpler maintenance, each module can be separately designed and optimized. The pipeline technique does, however, have significant drawbacks, including the requirement for explicit module design and the challenge of handling complicated and varied user inputs. The pipeline technique, which makes use of a number of modules including NLU, DST, Dialog Policy, and NLG, is a popular architecture for creating task-oriented dialog systems. For improved efficiency and maintenance, each module can be built and optimized independently. Each module in the dialog generating process is in charge of a particular duty. However, the pipeline approach also has some limitations and may not be suitable for all applications. This method can be difficult to optimize, though, as each module might have a unique set of parameters that need to be changed independently. Additionally, the pipeline approach's modules are very dependent on one another, which increases the risk of faults spreading throughout the entire system. Due to these problems, academics are now looking into other task-oriented dialogue system approaches, such as the end-to-end approach.

1.1.2 End-to-end approach

In recent years, the end-to-end method has grown in favor as a relatively new architecture for creating task-oriented dialogue systems. The end-to-end

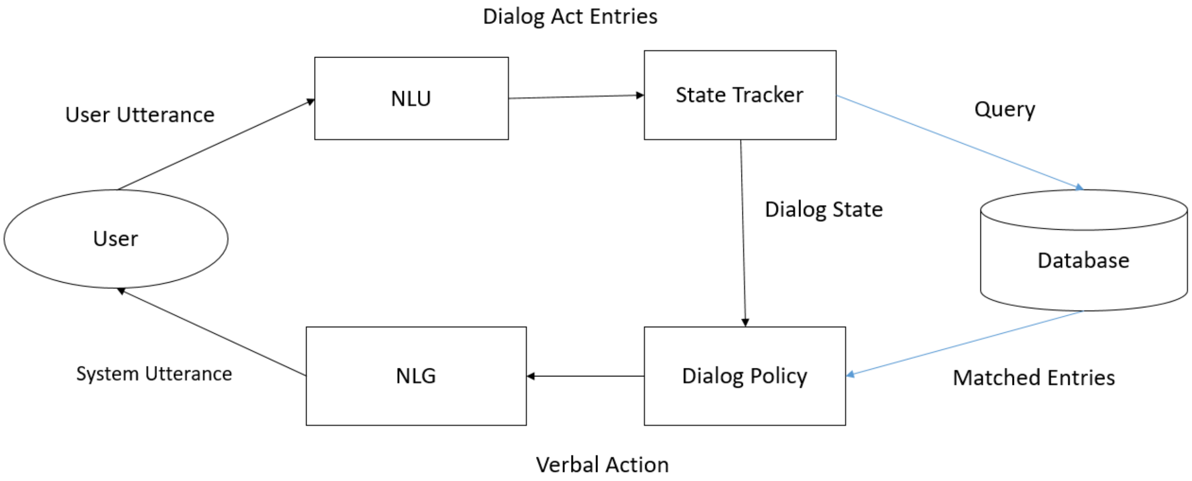


Figure 1.1: Architecture of The Pipeline Approach For Task Oriented Dialogue Systems

strategy makes use of a single neural network to produce responses to user input, in contrast to the conventional pipeline approach, which employs a number of modules to carry out various functions.

- **Architecture of the End-To-End Approach:** The end-to-end approach eliminates the need for intermediary representations like semantic frames or dialog actions by having the system take the user's natural language utterance as input and output the system's answer directly. Compared to the pipeline technique, which calls for the design and optimization of numerous distinct modules, the end-to-end approach is significantly more straightforward and easier to apply. Recurrent Neural Networks (RNNs) and Transformer models, which have been demonstrated to be particularly effective in natural language processing tasks like machine translation and text production, are the foundation of the end-to-end method. Large datasets of task-oriented dialogues that are annotated with the user's purpose and the system's response are used to train these models. The end-to-end approach's capacity to manage intricate and varied user inputs is one of its key benefits. The end-to-end technique can learn to manage a wide range of input changes from the training data, in contrast to the pipeline approach, which depends on explicit module design to handle different forms of user input. As a result, the end-to-end method is very scalable and flexible with regard to various domains and tasks. The end-to-end method's capacity to provide more fluid and natural reactions is another benefit. The end-to-end technique can produce replies that are more contextually appropriate and better reflect the nuanced aspects of natural language since it directly optimizes for the system's response rather than using intermediate representations. A more interesting and enjoyable user experience may result from this. The end-to-end strategy does, however, have significant drawbacks. The lack of interpretability and transparency is one of the major problems. It can be challenging to comprehend how the system creates its responses or to identify faults in the system's behavior because the model is a black box that directly transfers input to output. The system's maintenance and problem fixing may become difficult as a result. The necessity for a lot of training data presents another difficulty for the end-to-end method. Since the model directly optimizes for the system's reaction, good performance necessitates a substantial amount of high-quality annotated training data. This can be difficult in areas or for tasks where data collection is difficult or expensive. The end-to-end technique, which has various advantages over the conventional pipeline approach, is a promising architecture for creating task-oriented dialogue systems. It can produce more fluid and natural reactions and is simpler and more scalable. It does have certain drawbacks, though, such as the necessity for a substantial amount of training data and the lack of interpretability and transparency. Overall, the particular requirements and limitations of the application will determine which architecture is used, and both strategies have advantages and disadvantages.

- **T5 and LongT5 (Text To Text Transfer Transformer):** End-to-end task-oriented dialogue systems that make use of transformer models and encoder-decoder architecture have attracted increasing attention in recent years. The T5 (Text To Text Transfer Transformer) model, created by researchers like [1], is one of the most exciting advancements and the most advanced transformer in this field. The capacity of this model to provide excellent text output from a variety of input sources,

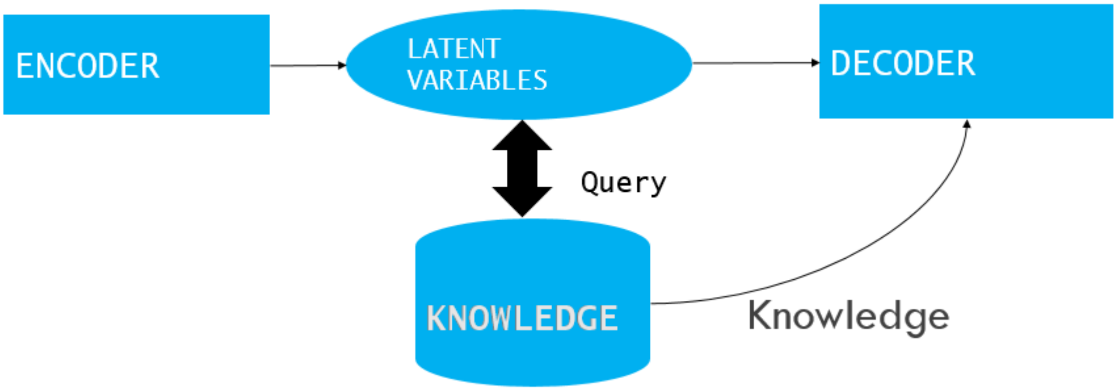


Figure 1.2: Architecture of The End-to-End Approach For Task Oriented Dialogue Systems

such as natural language searches and more structured data sources, makes it particularly noteworthy. The fact that the T5 model has been pre-trained on the enormous corpus of clean English text known as the C4 (Colossal Clean Crawled Corpus) dataset is one of its main advantages. This dataset, which is 750GB in size and contains text scraped from over 350 million web pages, is made up of hundreds of gigabytes of data. The T5 model has gained a wealth of information about a diverse variety of subjects and domains thanks to this pre-training procedure, which also helped it build a profound comprehension of the English language's structure and subtleties. Overall, the T5 model represents a significant advancement in the creation of end-to-end task-oriented dialogue system approaches. It is the perfect option for a wide range of applications and use cases due to its capacity to produce high-quality text output from a variety of input sources and its thorough pre-training on a sizable corpus of clean English text. In light of this, our research aims to advance this intriguing work by investigating the possibilities of a more modern transformer model known as LongT5, adapting it, and assessing its performance on a number of task-oriented discussion datasets, primarily MultiWoz 2.0, MultiWoz 2.1, and MultiWoz 2.2. A sophisticated transformer model called the T5 (Text To Text Transfer Transformer) has been pre-trained using the enormous dataset known as the C4 (Colossal Clean Crawled Corpus). The T5 model's self-attention mechanism, which enables it to focus on certain portions of the input text while processing it, is one of its important characteristics. In particular, the T5 model's self-attention mechanism employs a type of autoregressive attention, which limits the model's attention to previous outputs. Because it helps the T5 model produce high-quality output by taking into consideration the context of prior output tokens, this autoregressive attention mechanism is crucial. This strategy is especially well-suited for task-oriented dialogue systems, because the system must keep the conversation in a consistent context in order to comprehend and react to human input correctly. According to [1], the application of the T5 paradigm has already demonstrated considerable advances in the functionality of end-to-end task-oriented dialogue systems. This is due to the T5 model's ability to manage complicated input-output mappings and produce high-quality results across a variety of domains. The T5 model is already extremely effective, but there is still potential for improvement, especially in terms of the unique requirements of task-oriented dialogue systems. Because of this, the goal of our research is to examine the potential advantages of employing the newly proposed MegaT transformer model, which is based on the LongT5 architecture and is especially made to handle longer input sequences (up to 16384 tokens). With the use of this research, we intend to improve task-oriented dialogue systems' performance while also utilizing the most recent developments in transformer technology.

1.2 Attention Mechanisms

The attention mechanism is a key component of the transformer architecture, which is widely used in natural language processing tasks. In the LongT5 transformer model, specific types of attention mechanisms are used

1.2.1 Local Attention

Local attention is one type of the attention mechanism. This method modifies how the self-attention mechanism of regular attention operates such that each token in the input sequence can only pay attention to a specific number of its left- and right-hand surrounding tokens. These tokens are referred to as adjacent tokens since a token may attend to the same number of tokens on the right as it can on the left. Comparing this attention mechanism to the conventional method reveals several benefits. The attention operation is first made more efficient for processing longer input sequences by reducing its computational complexity. This attention mechanism's temporal complexity is mathematically $\mathcal{O}(n \times m)$, where n is the length of the input sequence and m is the radius, or the number of tokens to the left or right of each token that it can attend to. In comparison, the conventional method can be prohibitively expensive for larger sequences and has a temporal complexity of $mathcalO(n^2)$. Another advantage of local attention is its capacity to help models better capture the local dependencies between characters in the input sequence. By limiting each token to just considering a small number of nearby tokens, the model is forced to concentrate on the most important information for each token. In tasks like text generation or summarization, where the model must capture the most crucial data from the input sequence, this can be very crucial. Overall, the LongT5 transformer's local attention mechanism marks a significant development in the field of natural language processing. LongT5 is able to process longer input sequences more quickly and improve its capacity to capture local dependencies between tokens by making use of this attention technique. This could have significant implications for a wide range of applications, including question answering, text summarization, and machine translation.

1.2.2 Transient Global Attention:

An expansion of the local attention process is the idea of transient global attention. Transient global attention allows each token to attend to global tokens in addition to its surrounding tokens, whereas local attention only allows each token to focus on its immediate neighbors. Transient global attention enhances the model's ability to detect global dependencies in the input sequence in this manner. The input sequence is initially broken up into blocks of k tokens in order to implement transient global attention. The embeddings of the tokens included in each block are added up and normalized to provide a global token for each block. A series of global tokens are produced as a result of this procedure, and these tokens can be added to the attention mechanism as extra inputs. Transient global attention leads to a temporal complexity of $\mathcal{O}(n(m + n/k))$ in terms of computational complexity., where k is the size of the blocks used to obtain global tokens, m is the radius of the local attention window, and n is the length of the input sequence. Due to the additional calculations needed to calculate the global tokens, the introduction of transitory global attention raises the computational cost of the model in comparison to local attention. The advantage of paying more attention to global dependencies, however, might offset the additional computing expense. A novel strategy for improving the model's attention mechanism is the incorporation of transient global attention in the LongT5 transformer model. The model is better able to identify long-range relationships in the input sequence by allowing each token to care for both its local tokens and global tokens. But it's crucial to carefully weigh the trade-off between the advantages of global attention and the computational expense needed to put it into practice.

1.3 Wizard-of-Oz Datasets

Various domains of The Wizard of Oz The task-oriented dialogue collection known as MultiWoz contains a sizable number of written human-to-human discussions from many fields. It has been used in a number of research to train and assess task-oriented dialogue systems. It is a fully annotated dataset. MultiWoz 2.0, MultiWoz 2.1, and MultiWoz 2.2 are the three versions of the dataset; the latter version adds more domains and turns.The MultiWoz 2.0 and MultiWoz 2.1 datasets were used in studies by [1] to assess the effectiveness of their TOD system, which was created using the T5 model. On both datasets, their system was able to produce outstanding results, proving the value of using T5 throughout the entire process. In order to assess MegaT's performance on the MultiWoz datasets, we developed it using the LongT5 architecture and the transient global attention technique. Given that the LongT5 model is built to accommodate longer input sequences, we think it will produce better results. We aim to show the possibilities for additional advancements in task-oriented dialogue systems by contrasting the performance of our system with that of [1]. The Multi-Domain Wizard-of-Oz (MultiWoz) dataset is a useful tool for creating and evaluating task-oriented dialogue systems, in general. Our usage of the dataset will help MegaT and other ongoing research projects to enhance these systems' functionality and scalability.

1.4 Thesis Challenges

The performance of our models and the results of our tests were significantly impacted by a number of problems we ran into while conducting this research thesis. One of these issues was the small number and amount of datasets designed especially for task-oriented dialogue systems. There were not enough dialogues available for training and evaluation due to the dearth of these datasets. The computational expense involved in developing and refining large-scale transformer models like T5 presented another difficulty. These models required a lot of computing power and turned out to be computationally demanding. Additionally, because there was a 4000 tokens restriction on the task-oriented dialogue systems we used, they were unable to handle lengthy input sequences. Lastly, we encountered difficulties in defining appropriate evaluation metrics for dialog generation within task-oriented contexts. This aspect presented a challenge due to the unique nature of task-oriented dialogues.

1.5 Thesis Contributions

We proposed an adaptation of LongT5 in a new domain: task-oriented dialogue systems. This model, named **MegaT**, is able to handle long input sequences (up to 16000 tokens).
Also, we proposed the use of Transient Global Attention mechanism in such systems, to speed up the computation.

1.6 Thesis Outline

This book is organized as follows: Before discussing our unique method, we described the background studies in chapter 2, which are separated into dataset-based and system-based approaches. In the following chapter, chapter 3, we presented our suggested methodology with regard to the architecture of the new system MegaT by first giving an overview of the new architecture and then detailing how each module of the system operates and the procedure flow. In chapter 4, We initially presented the experimental setting and datasets before discussing the experimental design, the training and evaluation datasets, and How the T5 and MegaT models were trained, After that, we discussed Experiments and Results Analysis in chapter 5 Where we Show the results after we have conducted the experiments and we provide some clues about the results we got.

Chapter 2

Literature Review

Recent years have seen a substantial increase in interest in the development of conversational agents in the context of task-oriented dialogue systems. The creation of coherent and context-aware dialogues has shown promising outcomes when using a strong neural network model. This review of the literature focuses on the usage of datasets used in the literature for pre-training and fine-tuning as well as earlier task-oriented dialog systems to give an overview of the current research on dialog generation with conversational agents.

2.1 Datasets Used For Task-Oriented Dialog Systems

To train and assess model performance for dialog production using conversational agents, vast and diverse datasets are needed. The goal of this study of the literature is to give a general overview of the datasets frequently utilized in dialog generation studies. These datasets are essential for benchmarking and training dialog systems, allowing researchers to create models that provide responses that are coherent and appropriate for the given context.

2.1.1 Pre-training Dataset

One of the most popular datasets for pre-training transformer-based models like T5 is the Colossal Clean Crawled Corpus (C4) by [5], which is a huge dataset including billions of web pages. As one of the TensorFlow datasets, C4 is a freely accessible dataset. It has been employed to train n-gram language models, common sense reasoning models, and machine translation models. Due to its size and diversity, the C4 dataset is particularly appealing for pre-training NLP models. It is made up of crawled and cleansed web pages, creating a dataset that is largely devoid of noise and spam. The text on the web sites is extremely varied and includes news stories, social media posts, and product evaluations, offering pre-training models a rich supply of linguistic diversity. Numerous research have proved the value of using C4 as a pre-training dataset. [6] used the dataset to train a substantial n-gram language model, producing cutting-edge results on a number of benchmarks for language modeling. [7], it was shown that the dataset may be utilized to enhance the performance of machine translation by using C4 to mine parallel text.In the study [4], the authors employed C4 and a multi-task learning strategy to pre-train a transformer-based model for numerous NLP tasks. Task-oriented dialogue systems have also made use of C4 as a pre-training dataset. The MultiWOZ dataset, a sizable dataset of human-human task-oriented talks across several areas, was pre-trained using C4 by [4]. Pre-training on C4 has been proven to be successful in enhancing the model's performance on the MultiWOZ dataset. Briefly stated, the Colossal Clean Crawled Corpus (C4) is a sizable and varied dataset that has been extensively utilized for pre-training transformer-based models, including T5. A variety of NLP activities, including language modeling, machine translation, common sense reasoning, and task-oriented dialogue systems, have demonstrated it to be helpful in enhancing model performance. The accessibility of this dataset has facilitated recent developments in NLP and contributed to the development of pre-training as a key method for attaining cutting-edge performance.

2.1.2 Versions of MultiWoz Dataset

The MultiWoz (Wizard-Of-Oz) dataset is the most widely used dataset for training dialogue models. It was first introduced by [14], but modifications have been made to produce a more robust version of the dataset that is less sensitive to noise and free of some annotation errors. This resulted in the creation of MultiWoz 2.0 by Reference in 2018, an updated dataset that includes restaurant and attraction booking in addition to hotel booking.It is a more advanced version of MultiWoz 2.0 and has 10,438 dialogues as well as the same annotations as MultiWoz 2.1, which was released in 2019 by

[15]. It offers improved annotations and corrects some of the inconsistencies and mistakes seen in the earlier versions. There are still 10,438 dialogue occurrences, the same as in MultiWoz 2.0. A more trustworthy and consistent foundation for training and assessing dialogue systems is offered by the annotations in MultiWoz 2.1. By using this version of the dataset in conjunction with MultiWoz 2.2 by [16], which was released in 2021, researchers can create task-oriented discourse models that are more precise and durable. It further develops the annotations and builds on the advancements made in earlier versions. MultiWoz 2.2 adds dialogue-level annotations for dialogue success and task success in addition to the annotations provided in earlier versions. These additional comments enable a more thorough assessment of dialogue systems and offer information on the system's overall effectiveness. With the same number of dialogues as previous versions, MultiWoz 2.2 offers an updated and enhanced resource for developing and evaluating task-oriented dialogue systems.

2.2 Systems

Task-oriented dialogue systems are designed to help users do certain tasks, such booking a hotel or placing an order for food. These systems need to be able to comprehend user intentions, keep track of the discussion context, and produce suitable responses. Previous studies have investigated a range of strategies, including rule-based systems, statistical techniques, and more recently, models based on neural networks.

2.2.1 Previous Systems

Through a number of important studies, the field of task-oriented dialogue systems has made major strides in recent years. ”Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures” by [2] is one of the earliest and most important works. It introduced the end-to-end approach to task-oriented dialogue systems using a single sequence-to-sequence architecture with belief spans to track the dialogue belief states. This represented a substantial advancement over earlier methods that utilized pipeline topologies and distinct modules for various tasks. After this research, ”A Simple and Effective End-to-End Task Completion Dialogue System” by

[3] provided a system that took many relevant responses into account in a dialog setting, improving the naturalness and diversity of responses.

The authors investigated the use of multi-task learning and transfer learning from pre-trained language models to learn representations across multiple NLU tasks, leading to improved performance on a variety of benchmarks in ”Multi-Task Learning for Multi-Domain Task-Oriented Dialogue Systems” [4]. The T5 model, which employs a text-to-text strategy to address a variety of language challenges, was recently introduced in ”Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer” by [1]. The T5 model is extremely effective for jobs requiring natural language generation and understanding because it is built on the transformer architecture and has been pre-trained on a sizable corpus of English literature. A crucial method for getting cutting-edge performance on many Natural Language Processing (NLP) tasks is the pre-training of deep neural models on huge volumes of text data.

2.2.2 T5 pre-training Strategy

Using a method that substitutes masked tokens for spans of consecutive input tokens, the T5 model has been pre-trained. The model is trained to forecast the missing tokens in a sentence using a pre-training approach known as masked language modeling. A key stage in many natural language processing tasks, masked language modeling aims to assist the model to learn the contextual relationships between words in a sentence.

2.2.3 T5 attention mechanism

Relative location embeddings serve as the foundation for the model's self-attention mechanism in the case of T5. A sort of attention technique known as relative position embeddings enables the model to focus on tokens according to their relative positions in the input sequence. Long-range relationships between words are captured by this attention mechanism, which is crucial for tasks like language modeling, machine translation, and conversation creation. The self-attention mechanism of T5 uses relative position embeddings, which were first used in the field of natural language processing. For instance, [9] presented an attention mechanism that captures long-range interdependence between words by using relative position embeddings. Relative location embeddings were also used by [10] in their research on self-attentional methods for machine translation. The model can learn the contextual links between words in a sentence more successfully if it is given the freedom to pay attention to tokens based on their relative positions in the input sequence. This is especially helpful for jobs like dialogue generation and machine translation where the model must produce coherent and contextually relevant responses. Overall, the model's state-of-the-art performance on a variety of natural language processing tasks may be attributed to the employment of relative position embeddings and masked language modeling in the pre-training and self-attention mechanisms of T5, respectively. The attention mechanism employed in the T5 model is comparable to that suggested in the original transformer architecture by [8]. The T5 model is capable of generating high-quality text by using the self-attention mechanism to attend to all the input tokens and produce output tokens that are conditioned on the entire input sequence.

2.2.4 T5 for Task Oriented Dialogue Systems

Researchers have been investigating the usage of T5 in the creation of task-oriented dialogue systems recently. One noteworthy article is "Improving End-to-End Task-Oriented Dialogue Systems with A Simple Auxiliary Task" by [1], which outlines an end-to-end strategy for creating task-oriented dialogue systems using T5. The research suggests a straightforward auxiliary task for the T5-based dialogue system that entails anticipating the subsequent token in the response sequence. This task helps to boost the system's overall performance. In contrast to training individual components individually and then integrating them, Lee's study takes an end-to-end method that is based on the notion that the entire system should be trained simultaneously. This strategy enables the model to acquire a more comprehensive representation of the dialogue problem and provide responses that are more precise and coherent. The task-oriented dialogue system built using the T5 model and created by [1] demonstrates cutting-edge performance on various benchmark datasets, illustrating the utility of the T5 model in this field.

2.2.5 LongT5

In 2022, a paper titled ”LongT5: Efficient Text-To-Text Transformer for Long Sequences” was published by [11]. This paper introduced a new transformer model, LongT5, which is designed to handle long input sequences for language-related problems. The need for a specialized model capable of handling long sequences arises from the limitation of the original transformer model [8], which was built to handle a fixed-length input.LongT5 uses a pre-training strategy called PEGASUS, which was introduced by [Zhang et al., 2020a]. The PEGASUS strategy involves masking out key sentences in a document and asking the model to generate those sentences as a single string, effectively summarizing the document. This pre-training strategy has shown promising results in various natural language processing tasks, including text summarization and language understanding.The LongT5 model builds on the PEGASUS pre-training strategy, with particular inspiration from the work done by [12]. This strategy enables the LongT5 model to handle longer input sequences by compressing the input text into a shorter summary, which is then used as the input to the model. This approach allows the model to learn more efficiently and effectively, as it focuses only on the key information in the input sequence, rather than being overwhelmed by irrelevant details.In the context of task-oriented dialogue systems, the LongT5 model has the potential to improve performance by allowing for longer input sequences and more comprehensive understanding of the user’s intent. The ability to handle longer sequences also enables the model to incorporate more context into the conversation, improving the system’s ability to respond appropriately to the user’s needs.The LongT5 model represents a significant step forward in the development of transformer-based models for language-related problems. By incorporating the PEGASUS pre-training strategy and optimizing for long input sequences, the LongT5 model has the potential to significantly improve the performance of task-oriented dialogue systems and other natural language processing applications.The data set used for pre-training is the C4 by [5]. The transient global attention has been inspired by ETC’s local/global mechanism in [13].

2.2.6 Local and Transient Global Attention

The first transformer model was proposed by [8] and makes use of a self-attention mechanism that enables each token in the input sequence to pay attention to every other token in the sequence. However, this method's temporal complexity is quadratic in respect to the length of the input sequence. By combining LongT5 and transitory global attention, which was inspired by ETC's local/global method in [13], MegaT overcomes this restriction.Similar to the original attention method, local attention in LongT5 only allows for attention to be focused on a narrower window of tokens surrounding the current token. This makes computing more effective and solves the quadratic growth problem. On the other hand, transient global attention expands on the concept of local attention by enabling each token to attend to global tokens in addition to nearby tokens. By totaling and normalizing the token embeddings within a block, this global focus is attained.The local/global method in ETC's [13] that uses a two-level attention mechanism to overcome the problem of transformers' quadratic computation growth served as the model for transitory global attention. This approach applies global attention to tokens in a bigger window around the local window, while local attention is applied to tokens in a small window around the present token. This strategy enables the model to keep a tolerable time complexity while attending to global information.

2.3 Limitations of Existing Systems and Datasets

[1] has worked on MultiWoz 2.0 and 2.1 in their paper and explores the use of an auxiliary task, span prediction in their case where by taking a span of input tokens They try to predict the attribute of the domain with the goal of enhancing the overall performance of the system. The similar research was done in the paper "Improving Pre-training by Representing and Predicting Spans" by [17].There are particular issues that need to be resolved in the end-to-end method, when a conversation system generates responses directly without relying on predetermined dialogue acts or slot-value pairings. End-to-end task-oriented dialogue systems face a number of difficulties, such as:

- Data Scarcity: It can be difficult to gather a significant amount of labeled training data for end-to-end dialogue systems. Gathering human-generated conversations for training end-to-end systems in multiple domains and scenarios is time- and resource-intensive compared to conventional slot-filling methods.

- Systematic Errors: End-to-end dialogue systems are prone to systemic errors, which occur when they repeatedly produce inaccurate or biased responses. These mistakes may result from biases in the training data, a dearth of examples from other backgrounds, or restrictions on the model's capacity to generalize outside of the training set.

- Lack of Control and Interpretabilty: End-to-end dialogue systems may lack explicit control over the generated responses, making it challenging to make sure the system complies with certain restrictions or adheres to desired guidelines. Additionally, it is difficult to comprehend and articulate how the system decides to act due to the underlying model's lack of interpretability.

- Context and Coherence: It might be difficult for end-to-end systems to keep context and produce coherent responses across a multi-turn debate. For the model to produce pertinent and contextually acceptable responses, it must comprehend and remember the context from prior rounds.

- Open-domain Versatility: While end-to-end dialogue systems thrive in certain task-oriented domains, they may have trouble handling conversations across open domains or responding to inquiries that fall outside of their training area. In end-to-end systems, handling out-of-domain requests and giving graceful fallback responses are challenging.

- Computational Resources: Large-scale neural network designs, for example, complex models, require a lot of processing power to train and deploy. It can be time-consuming and computationally expensive to train these models, especially when dealing with complex dialogue scenarios or enormous volumes of data. Real-time systems may also need a lot of computational resources to implement these models.

- Data Requirements: More complex models often have higher data requirements for training. They may require larger and more diverse datasets to learn complex patterns and generalize well to different dialogue scenarios. Collecting and curating such datasets can be challenging, particularly for specialized domains or rare dialogue types.

Chapter 3

Proposed Methodology

The proposed methodology aims to develop a dialog generation system using a new Transformer architecture in the context of task-oriented dialogue named MegaT. The system will generate responses directly without relying on predefined dialogue acts or slot-value pairs.

3.1 Overview

Here is an overview of our proposed Architecture diagram ,The methodology consists of several key components: a dialogue encoder, a belief decoder, a database, a response decoder.

3.2 Dialogue Input

As shown in Fig 1 ([1]), the System takes a sequence as input and outputs a sequence. A Conversation consists of multiple turns where the user and the system utter interchangeably. During the conversational system's turn, the encoder within the system takes the user utterance Ut and past conversations H_t as inputs, past conversations H_t consist of sequence of tuples where each tuple contains a user utterance ,a belief state,a database state,a system action and a response in a previous time step.

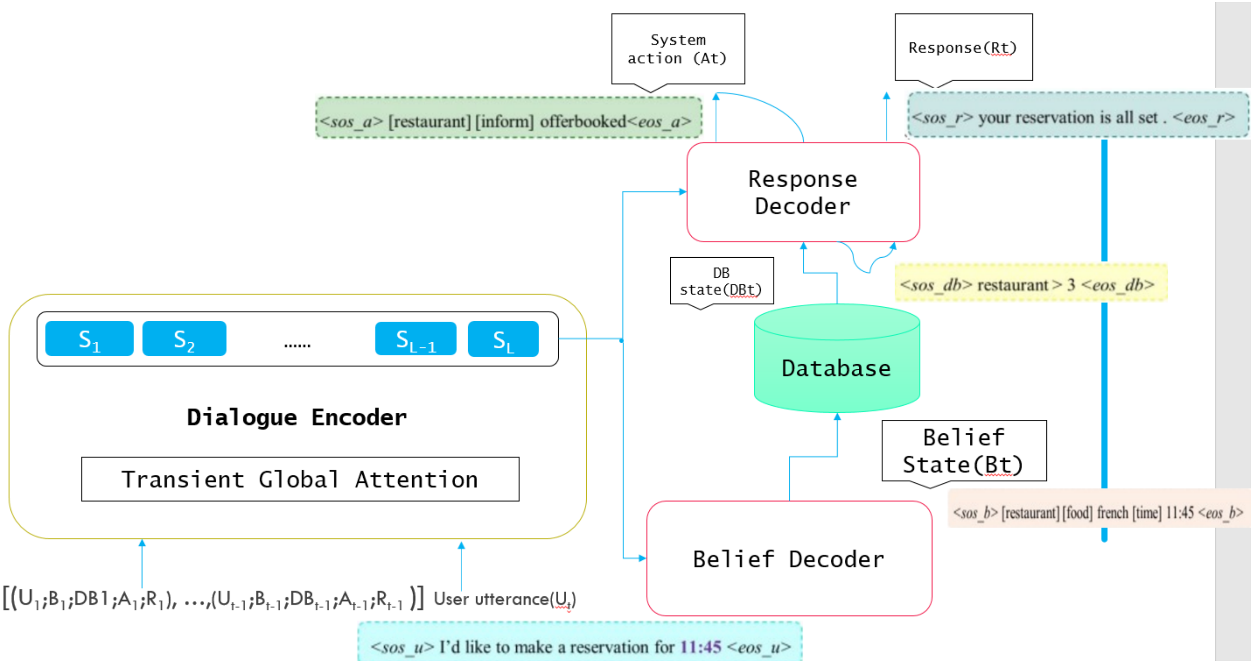

System action (At)
Response(Rt)
<sos_a> [restaurant] [inform] offerbooked<eos_a>
<sos_r> your reservation is all set . <eos_r>
Response Decoder
DB state(DBt)
<sos_db> restaurant > 3 <eos_db>
S_1
S_2
......
S_{L-1}
S_L
Database
Dialogue Encoder
Belief State(Bt)
Transient Global Attention
<sos_b> [restaurant] [food] french [time] 11:45 <eos_b>
Belief Decoder
$[(U_1;B_1;DB1;A_1;R_1), \ldots, (U_{t-1};B_{t-1};DB_{t-1};A_{t-1};R_{t-1})]$ User utterance(U_t)
<sos_u> I'd like to make a reservation for 11:45 <eos_u>

Figure 3.1: An Overview of The Conversational Agent System

3.3 Dialogue Encoder

A representation encoded by the dialogue encoder is created. An easier form of the input data for the system to handle and analyze is the encoded representation, which is simply a condensed and meaningful version of the input data. In order for the system to produce a meaningful response, it needs to be able to recognize key elements and patterns in the input data that are captured by the encoded representation.

3.4 Belief Decoder

Following that, the belief decoder, which creates a belief state B_t, receives the encoded representation. The belief state is a 3-tuple (domain, slot, value), which communicates the topic, attributes, and values of the dialogue as it stands right now. The domain denotes the overall subject of the discussion, the slot denotes a particular attribute connected to that domain, and the value denotes the value of that slot. For instance, the domain, slot, and value in a restaurant reservation system might all be "restaurant," "time," and "7pm," respectively. The belief state is crucial because it guides the system's subsequent actions and reactions. The system can better decide what information to deliver and what steps to take by comprehending the conversation's current situation. To ensure that the system has a current and correct understanding of the dialogue, the belief state is also updated as the discussion goes on.The encoder and belief decoder are crucial parts of the conversational system because they allow it to analyze and comprehend user input and produce responses that are relevant to the conversation at hand. While the belief state collects crucial details about the current status of the discussion in terms of the topic, attributes, and values, the encoded representation offers a reduced and comprehensible version of the input data.A domain-defined database—a collection of material that is pertinent to the conversation's current topic—is queried using the belief state. For instance, the database for a restaurant reservation system can contain details on the menu items, available tables, and reservation hours. The information from the database that is pertinent to the present stage of the dialogue is selected using the belief state.

3.5 Database

he domain-defined database's current state is represented by the DB state DB_t. The slots and values in the belief state are used to calculate the number of matching units in the database, which is used to update the database state. The DB state would be modified to reflect the tables that were accessible at that particular time, for instance, if the belief state contains a value for the time slot.The DB state is significant because it gives the system the data it requires to produce intelligent responses and actions. The system can give the user accurate and pertinent information and be able to take suitable actions based on the user's input by querying the database and updating the DB state.Based on the encoded representation and DB status, the response decoder is essential in creating a system action. The system action is a 3-tuple (domain, action type, and slot), where the slot is the attribute of the domain that is to be acted upon and the domain is the topic of the conversation.

3.6 Response Decoder

The response decoder generates a system action from the input of the encoded representation and DB state. Based on the current belief state, it generates a list of potential system actions using the encoded representation. Each potential course of action is evaluated for its applicability to the subject under discussion, and the resulting scores are used to rank the possible courses of action.The candidate acts that are inappropriate for the current discourse are filtered out using the DB state. For instance, the system should only recommend travel-related actions, such as picking a date, choosing a location, and choosing a preferred airline, if the topic of conversation is booking a flight.Following filtering, the remaining potential actions are scored for relevance, and the action with the highest score is chosen as the system action. In order to provide a natural language answer pertinent to the current conversation, this action is then input back into the response decoder.The system action is mapped to a predefined set of answer templates to produce the natural language response. Each response template is intended to produce a response that is appropriate for the present conversation and correlates to a certain system activity. For instance, the appropriate answer template would be customized to produce a response that proposes a suitable date if the system action was to suggest a date for a flight.A crucial part of conversational agents is the response decoder, which enables the system to provide pertinent and fitting responses based on the conversation's present state. The response decoder may efficiently provide system actions that are in line with the user's demands and preferences by merging the encoded representation and DB state, resulting in a more natural and interesting discussion.

3.7 Procedure Flow

The current procedure flow $(U_t; B_t; DB_t; A_t; R_t)$ is added to the previous conversations H_{t+1} till the conversation is over. By taking into account the discrepancy between the anticipated and actual belief states and system actions/responses, the belief and system action/response loss functions are defined. Using backpropagation, the parameters of the belief and response decoder are trained to reduce the loss functions.The encoder uses an LSTM network to convert the user's input Ut, which consists of a series of words or sentences, to a fixed-length encoded representation in the belief state generation process. The belief decoder, a multi-label classification model, is then fed the encoded representation. The encoded representation and the database state are concatenated, and the resulting vector is used as input to the system action decoder. The system action decoder is a multi-label classification model that outputs a probability distribution over all possible system actions.Finally, the response generation process takes the encoded representation, the database state DB_t, and the system action At as input. The encoded representation and the database state are concatenated, and the resulting vector is used as input to the response decoder. The response decoder is a sequence-to-sequence model that generates the natural language response Rt.In summary, the dialogue system employs an encoder-decoder architecture to generate a belief state, query a domain-defined database, generate a system action, and produce a natural language response. The belief and response decoders are trained using a backpropagation-based algorithm that minimizes the difference between the predicted and actual belief states and system actions/responses

The belief and system action/Response loss functions are defined by

$$\mathcal{L}_{belief} = -\ log\ p(B_t|H_t, U_t), \qquad (1)$$

$$\mathcal{L}_{resp} = -\ log\ p(A_t, R_t|H_t, U_t, DB_t), \qquad (2)$$

This loss function measures the accuracy of the system action and response.

This system is designed to facilitate conversation between a user and a system by processing input data, generating a belief state, querying a database, and producing a natural language response. The performance of the system is evaluated using loss functions that measure the accuracy of the belief state and the system action/response.

Chapter 4

Experimental Design

To evaluate the performance of the proposed dialog generation system in the context of task-oriented dialogue using a Transformer architecture, an experimental design is proposed. The design consists of the following components: baseline comparison, evaluation metrics, experimental setup, and data analysis.

4.1 Environment Setup

The training and evaluation phases of our project were performed on Google Colab Pro Plus, a cloud-based development environment that offers several features and benefits. This platform provides users with 500 compute units per month, which expire after a period of 90 days. If additional compute units are required, users can purchase them as needed. One of the key advantages of Google Colab Pro Plus is its faster GPUs (A100 GPU), which are optimized for performance and enable users to train and evaluate complex machine learning models more quickly and efficiently.Another benefit of Google Colab Pro Plus is its priority access to upgrade to more powerful premium GPUs. This allows users to access the most powerful GPU resources available on the platform, which can significantly accelerate the training and evaluation of deep learning models. In addition, Google Colab Pro Plus offers 52 GB of RAM, which is a substantial amount of memory that can support the processing and manipulation of large datasets. with a Batch size of 8 , Adafactor optimizer , a Learning rate of 0.001 and a Dropout rate of 0.1 .Furthermore, Google Colab Pro Plus allows for background execution, which means that users can continue to run tasks even when their browser windows are closed. This feature is particularly useful when training and evaluating models that require long execution times or when working on projects that require extensive processing power.

4.2 Datasets Used For The Experiment

MultiWoz is a useful dataset that was compiled using the wizard-of-oz technique, in which consumers were misled into thinking they were corresponding with computers when, in fact, they were speaking with real wizards. Natural language processing (NLP) models that can precisely comprehend and react to user input can be developed using the data acquired through this approach, which is particularly helpful for training conversational AI systems.Seven domains are included in the MultiWoz dataset: hotel, train, attraction, restaurant, taxi, hospital, and police. Users can input a variety of data within each domain, including the name of the hotel, the check-in and check-out dates, the number of guests, and the restaurant's preferred cuisine. These pieces of information are represented as 16 slots, which are specific data fields that NLP models can learn to recognize and extract from user utterances.

4.2.1 Statistical Description of The Datasets

A popular dataset for task-oriented discussion systems is MultiWoz 2.0. There are 10,438 dialogues total in it, with 7 different domains (hotel, restaurant, attraction, train, taxi, hospital, and police) represented. It provides a wide variety of interactions throughout the course of 115,424 turns. The dataset has 47,009 different slot values, which enables models to respond to a range of user requests. Notably, a high task success rate of 95% of the conversations results in the user's desired outcome. This version is a useful tool for developing and assessing task-oriented discussion systems.

Dialog act annotations are a new feature in MultiWoz 2.1, which improves upon MultiWoz 2.0. The 10,438 talks, 7 domains, 115,424 turns, and 47,009 different slot values from the previous version are all still present. However, the addition of dialogue act annotations enables model developers to examine and enhance system performance by giving them a more detailed understanding of the conversation structure. The dataset's usability and richness are both improved by the addition of this annotation.

A development of MultiWoz 2.0 and 2.1, MultiWoz 2.2 adds more system dialogue act annotations. 10,438 talks, 7 domains, 115,424 turns, and 47,009 different slot values are all present in the collection. An improved knowledge of system reactions and behavior is made possible by the system dialogue act annotations, allowing for more sophisticated dialogue modeling and system evaluation. This version enables researchers and developers to investigate interactions at the system level and investigate fresh methods for conversation system development.

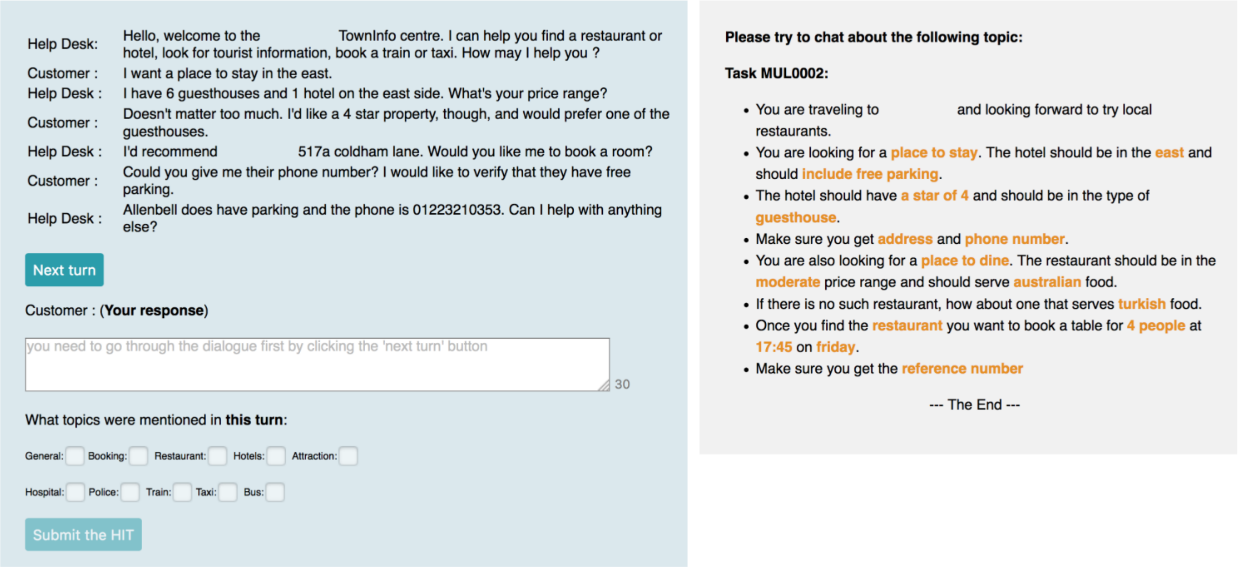
The diversity of MultiWoz is another benefit. The dataset includes a wide range of events and issues that users may run into when interacting with companies and services in the real world. Users can inquire about the availability of hotel rooms, reserve a table at a restaurant, or look for the location of the closest hospital, for example. We can guarantee that these systems can handle a wide range of use scenarios and offer useful responses to users by training AI models on such a diversified set of data.Additionally, the MultiWoz dataset contains extensive semantic annotations that make it simpler for researchers to assess the effectiveness of their models. The corresponding slot values, intent, and dialogue act are identified for each user statement and system response. These labels give researchers the ability to evaluate how accurately their algorithms identify and extract pertinent data from user input, produce appropriate responses, and conduct meaningful conversations.For the advancement of NLP and the creation of conversational AI systems that can effectively understand and respond to user input, the MultiWoz dataset is a crucial resource. It serves as the perfect baseline for comparing the effectiveness of various models and methodologies thanks to its size, diversity, and thorough annotation. We can pave the road for more sophisticated and successful conversational agents that can assist users in navigating challenging activities and interacting with businesses and services more effectively by continuing to enhance and grow the MultiWoz dataset.

4.2.2 Dataset Selection

The Wizard-of-Oz (WoZ) technique was used to collect a fraction of the dialogues in the MultiWOZ dataset. In the dialogue interaction of the Wizard-of-Oz collection method, a human "wizard" simulates both the user and the system. By utilizing human knowledge, this technology aims to record more complicated and organic discussions.The following steps are commonly included in the Wizard-of-Oz method gathering process:

- Designing Scenarios: Based on the target domain or application, scenarios or tasks are defined. The precise tasks or objectives that users want to accomplish through dialogue interaction are represented by these scenarios.
- Wizard and User Roles: A human "wizard" is in charge of acting in the capacities of the user and the system. Another human player who plays the part of the user converses with the wizard. Based on the user's input, the wizard has access to a set of predefined responses or actions from which to choose.
- Dialogue Interaction: The user and the wizard have a conversation in which the user makes requests and the wizard responds appropriately. The wizard generates the proper system answers or actions using their knowledge and skills.
- Natural Language Generation: Based on the user's input, the wizard creates system responses in natural language. These responses ought to be in line with the discussion's objectives and give the user information that is pertinent and coherent..
- Recording and Annotation: The user and wizard's conversations are taped and annotated. In addition to slot-value pairs that record the crucial information shared during the conversations, the annotations may also include dialogue acts, which indicate the intentions or actions of each user turn..

As human wizards are able to give nuanced and contextually relevant responses, the Wizard-of-Oz technique enables for more realistic and varied dialogues in the collection of the MultiWOZ dataset. This method improves the quality and depth of the dataset while capturing the complexity of natural language exchanges in task-oriented dialogue systems.



Help Desk: Hello, welcome to the TownInfo centre. I can help you find a restaurant or hotel, look for tourist information, book a train or taxi. How may I help you ?

Customer : I want a place to stay in the east.

Help Desk : I have 6 guesthouses and 1 hotel on the east side. What's your price range?

Customer : Doesn't matter too much. I'd like a 4 star property, though, and would prefer one of the guesthouses.

Help Desk : I'd recommend 517a coldham lane. Would you like me to book a room?

Customer : Could you give me their phone number? I would like to verify that they have free parking.

Help Desk : Allenbell does have parking and the phone is 01223210353. Can I help with anything else?

Customer : (**Your response**)

you need to go through the dialogue first by clicking the 'next turn' button

30

What topics were mentioned in **this turn**:

General: □ Booking: □ Restaurant: □ Hotels: □ Attraction: □

Hospital: □ Police: □ Train: □ Taxi: □ Bus: □

Please try to chat about the following topic:

Task MUL0002:

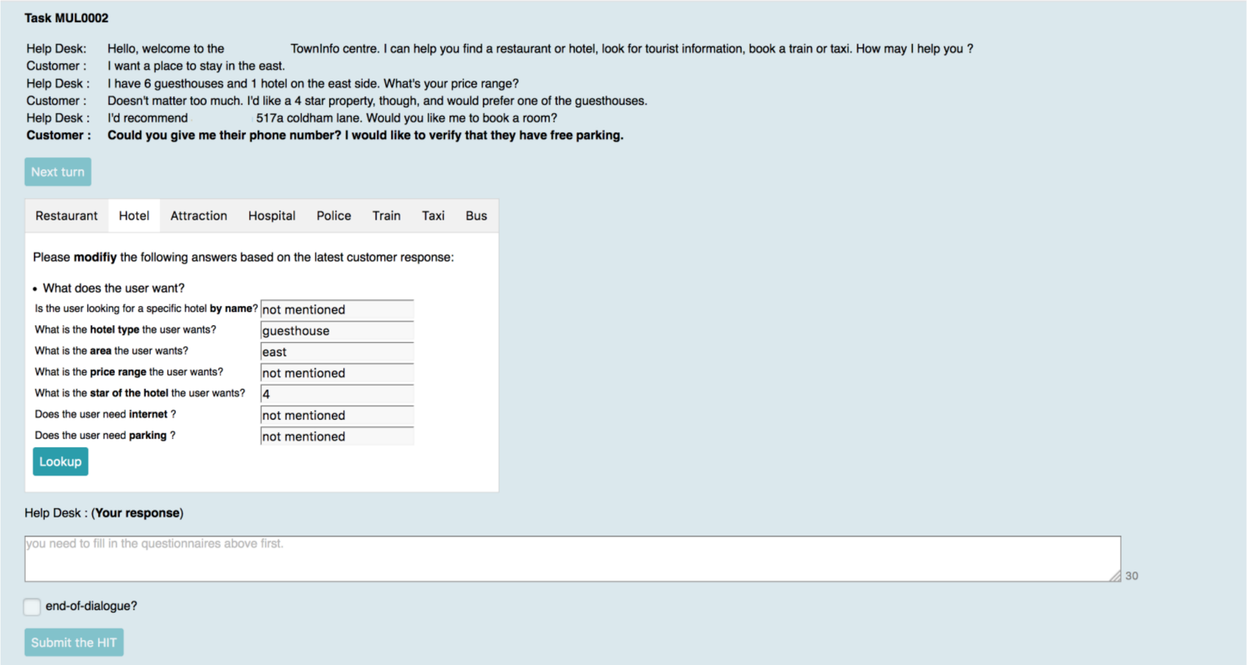
- You are traveling to and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the **east** and should **include free parking**.
- The hotel should have **a star of 4** and should be in the type of **guesthouse**.
- Make sure you get **address** and **phone number**.
- You are also looking for a **place to dine**. The restaurant should be in the **moderate** price range and should serve **australian** food.
- If there is no such restaurant, how about one that serves **turkish** food.
- Once you find the **restaurant** you want to book a table for **4 people** at **17:45** on **friday**.
- Make sure you get the **reference number**

--- The End ---

Figure 4.1: User Portal adapted from [14]

4.3 Model Training

We wrote our codes on Visual Studio Code. In order to train the two models T5 and MegaT in the Google Colab environment, we uploaded the codes and datasets in the Google Drive , then made a connection between Google Drive and Google Colab Environment to access our codes and the datasets and we install some dependencies that we needed to run the codes. As the two models can be train in a command line interface we just wrote the necessary commands each time to train the models. for each model , the code reside in a specific folder, which consists of multiple files



Task MUL0002

Help Desk: Hello, welcome to the TownInfo centre. I can help you find a restaurant or hotel, look for tourist information, book a train or taxi. How may I help you ?
Customer : I want a place to stay in the east.
Help Desk : I have 6 guesthouses and 1 hotel on the east side. What's your price range?
Customer : Doesn't matter too much. I'd like a 4 star property, though, and would prefer one of the guesthouses.
Help Desk : I'd recommend 517a coldham lane. Would you like me to book a room?
Customer : Could you give me their phone number? I would like to verify that they have free parking.

Please **modifiy** the following answers based on the latest customer response:

- What does the user want?

Is the user looking for a specific hotel **by name**?	not mentioned
What is the **hotel type** the user wants?	guesthouse
What is the **area** the user wants?	east
What is the **price range** the user wants?	not mentioned
What is the **star of the hotel** the user wants?	4
Does the user need **internet** ?	not mentioned
Does the user need **parking** ?	not mentioned

Help Desk : (**Your response**)

you need to fill in the questionnaires above first.

□ end-of-dialogue?

Figure 4.2: Wizard Portal adapted from [14]

Chapter 5

Experiments And Results Analysis

In our study, we evaluated the performance of the old TOD system implemented with T5 model and the new TOD system MegaT on the MultiWoz 2.0 , MultiWoz 2.1 and MultiWoz 2.2 datasets. We aimed to compare the performance of the two systems on various metrics, namely belief accuracy, belief loss, response loss, and response accuracy.

5.1 Evaluation Metrics

5.1.1 Belief accuracy

The Belief Accuracy is a crucial metric that evaluates the performance of the belief decoder. It measures how well the outputted belief state by the belief decoder corresponds to the actual belief state of the dialogue. A high belief accuracy indicates that the system can accurately interpret the user's intents and update the belief state accordingly.

5.1.2 Belief loss

The Belief Loss is another metric that is measured as the cross-entropy loss between the predicted belief state by the belief decoder and the actual belief state of the dialogue. A low belief loss indicates that the system can accurately predict the user's intents, and the belief state remains consistent with the user's goals and preferences throughout the dialogue.

5.1.3 The response loss

The Response Loss is the cross-entropy loss between the predicted system action by the Response Decoder and the correct system action. The response decoder generates the system's response based on the current dialogue state and the user's intents. A low response loss indicates that the system can generate appropriate responses that are relevant to the user's intents and goals.

5.1.4 Response accuracy

It determines how good the predicted system action corresponds to the correct system action. It evaluates the system's ability to generate the correct response that is relevant to the user's intents and goals. A high response accuracy indicates that the system can generate appropriate responses that satisfy the user's needs and preferences.

5.2 Results Analysis

The primary objective of our research was to assess and contrast the effectiveness of two task-oriented dialogue (TOD) systems, namely the T5 system and the MegaT system, using the MultiWoz 2.0, MultiWoz 2.1, and MultiWoz 2.2 datasets. We sought to evaluate these systems based on the specified metrics, aiming to provide a comprehensive analysis of their performance in the context of task-oriented dialogue generation.

	Belief Loss	Belief Accuracy	Response Loss	Response Accuracy
T5	820.76	98.67	742.05	78.72
MegaT	743.53	98.75	687.14	79.11
Improvement	-77.23	+0.08	-54.91	+0.39

Table 5.1: **Comparison of T5 and MegaT on MultiWoz 2.0**

	Belief Loss	Belief Accuracy	Response Loss	Response Accuracy
T5	833.69	98.75	756.85	78.45
MegaT	758.22	98.85	703.52	78.80
Improvement	-75.47	+0.1	-53.33	+0.35

Table 5.2: **Comparison of T5 and MegaT on Multiwoz 2.1**

	Belief Loss	Belief Accuracy	Response Loss	Response Accuracy
T5	782.70	98.84	712.21	79.76
MegaT	710.05	98.99	662.67	80.19
Improvement	-72.65	+0.15	-49.54	+0.43

Table 5.3: **Comparison of T5 and MegaT on Multiwoz 2.2**

	Belief Loss	Belief Accuracy	Response Loss	Response Accuracy
Local Attention	750.09	98.94	696.39	79.87
Tglobal Attention	710.05	98.99	662.67	80.19
Improvement	-40.04	+0.05	-33.72	+0.32

Table 5.4: **MegaT Local Attention vs Transient Global Attention on MultiWoz 2.2**

5.2.1 Analysis

Our results showed that the new proposed TOD system MegaT outperformed the T5 system on all the metrics evaluated.T he belief accuracy and response accuracy of the new TOD system MegaT were significantly higher than the other systems.

The results above show an increase in the Belief and Response accuracy of our proposed TOD system MegaT Based on LongT5‘over the previous TOD system T5. We also observe a decrease in the Belief and Response losses. During Our Experiment we have observed that the use of Adafactor Optimizer considerably improve the memory usage during Training and the use of other Optimizer like Adam or RMSProp optimizers still Lead to good performance but consumes a lot of memory during training , Using a learning rate of less that 5e-4 slows down the learning process of both MegaT and T5 and the training phase takes more time , and the use of higher learning rates lead to bad performances. The results obtained on MultiWoz 2.2 demonstrate superior performance compared to the previous versions, MultiWoz 2.0 and MultiWoz 2.1. The performance of the dialogue system has improved significantly across the board, according to the evaluation criteria and benchmarks used to measure it. These enhancements are attributable to MultiWoz 2.2's inclusion of system conversation act annotations, which offer a more thorough understanding of system replies and behavior.MultiWoz 2.2 supports more sophisticated dialogue modeling methodologies and system evaluation thanks to the addition of system dialogue act annotations. By more properly capturing the system-level interactions, the underlying discourse dynamics may be understood. Researchers and developers can create more effective interaction methods and provide more contextually relevant and coherent system responses because to this improved understanding of system behavior.The enhanced MultiWoz 2.2 performance demonstrates the beneficial effects of utilizing system dialogue act annotations in task-oriented dialogue systems. It highlights how models can more closely match user requests and deliver more precise and satisfying responses by taking into account the specific actions and purposes of the dialogue system. These findings emphasize how crucial it is to include fine-grained annotations in dialogue datasets in order to progress dialogue system development's state-of-the-art.

The results on MultiWoz 2.2 show that the model MegaT with Transient Global Attention outperforms MegaT with Local Attention in terms of Belief Loss, Belief Accuracy, Response Loss, and Response Accuracy, among other assessment criteria.The Belief Loss metric calculates the difference between the dialogue system's ground truth belief states and predictions. Lower Belief Loss for MegaT with Transient Global Attention demonstrates MegaT's capacity to more accurately collect and model the system's real belief states. The model with transient global attention may thus be better able to comprehend and reflect the user's intentions and preferences throughout the conversation.The Belief Accuracy metric evaluates the accuracy of the predicted belief states in a similar manner. In terms of Belief Accuracy, MegaT with Transient Global Attention performs better than MegaT with Local Attention, demonstrating that the former is better at correctly forecasting the system's beliefs based on the dialogue context. This suggests a greater comprehension of the user's demands and expectations, resulting in responses that are more accurate and contextually suitable.The difference between the generated and reference replies is measured by the Response Loss, and MegaT with Transient Global Attention yields a smaller loss value. This shows that the responses this model generates are more similar to the reference responses, meaning that they are of higher quality and coherence. The discussion history's contextual information can be successfully captured and incorporated by the model with transient global attention, resulting in more thoughtful and appropriately contextualized responses.The Response Accuracy statistic additionally assesses the accuracy of the generated responses. In comparison to MegaT with Local Attention, MegaT with Transient Global Attention exhibits greater Response Accuracy, demonstrating the latter's capacity to provide more correct and contextually relevant responses. This shows that the transient global attention model understands the dialogue context better and may produce responses that are well-aligned with the user's requests and preferences.Overall, MegaT with Transient Global Attention outperformed other models in these assessment criteria, demonstrating how well it can represent dialogue systems for task-oriented dialogues on the MultiWoz 2.2 dataset. The model can better capture long-term dependencies and contextual information thanks to the transient global attention mechanism, which leads to more accurate belief monitoring and the production of high-quality replies.

Chapter 6

Conclusion and Future Works

It has become a focus of research to create conversational agents for task-oriented dialogue systems since it holds great promise. The end-to-end method has proven to be exceptionally stable and effective, providing a workable alternative for creating task-oriented conversations. The development of scalable models that can handle long input lengths has been made easier by the introduction of transformer models. Our main goal in this study was to assess the effectiveness of two task-oriented dialogue systems, T5 and MegaT (a new TOD system built on LongT5), and show MegaT to be superior.We experimented with the MultiWoz 2.0, MultiWoz 2.1, and MultiWoz 2.2 datasets to achieve this. According to our findings, MegaT consistently outperformed T5 in all of the evaluation measures taken into account. In comparison to current state-of-the-art models, MegaT demonstrated gains in belief accuracy, belief loss, response accuracy, and response loss, demonstrating its superiority in creating task-oriented conversations. This comparison not only highlights MegaT's effectiveness but also sheds important light on the models' comprehension of user goals, production of precise responses, and dialogue coherence.The study's research has highlighted a number of issues and topics within the subject of task-oriented dialogue systems that need more study and inquiry. These challenges represent opportunities for researchers to advance the capabilities of conversational agents in task-oriented dialogues and enhance their ability to understand and respond to user needs more effectively.

- Transformer architecture tuning is a crucial topic that requires focus.

Transformers have shown impressive performance in natural language processing tasks, but they can be further improved by being fine-tuned specifically for task-oriented dialogue systems. The model's capacity to handle certain dialogue tasks can be optimized by researchers by fine-tuning the architecture, leading to more precise and context-aware responses.

- The development of task-oriented dialogue systems has a lot of potential with transfer learning approaches as well. The performance of conversational agents can be greatly improved by utilizing pre-trained models on extensive language problems and applying their knowledge to task-oriented dialogues. This method enables models to take advantage of the comprehensive knowledge captured during pre-training while adjusting to the unique dialogue setting.

- Approaches to reinforcement learning provide yet another way to advance. Conversational agents can learn by making mistakes and gradually improve their dialog methods by using reinforcement learning algorithms. Reinforcement learning can aid agents in navigating convoluted conversations, successfully handling user demands, and giving responses that are more agreeable.

- Further research is needed in another crucial area: multi-turn and contextual awareness. Dialogues are naturally dynamic and frequently entail several turns, necessitating that agents keep context and comprehend the current discourse. The effectiveness of task-oriented dialogues and the user experience can both be considerably increased by improving models' capacity to track discussion history, infer user intent, and create coherent responses across numerous rounds. Another crucial element that can be improved is user modeling.

- Conversational agents can customize their responses and the interaction to each user by capturing and modeling the user's preferences, objectives, and traits. Agents are able to offer recommendations or assistance that are more pertinent and personalized by taking into account the user's profile, history, and interests.

- Another topic for future research is real-time interaction. The user experience can be improved and more engaging and natural interactions can result from giving conversational bots the ability to participate in real-time, interactive dialogues. Agents must process and react to user inputs in real-time while taking into account both the status of the dialogue at the time and the changing context.

Researchers can push the bounds of task-oriented dialogue systems and open up new possibilities for conversational bots by solving these issues and further researching these topics. As a result, conversation systems will become more complex and productive and be able to recognize and respond to user needs more effectively. This will enhance the user experience overall and allow for more natural human-computer interactions.By contrasting the performance of T5 and MegaT on the MultiWoz datasets, this study contributes to the field of task-oriented dialogue systems. Our findings demonstrate MegaT's superiority and its capacity to outperform other models in the development of task-oriented discourse. The evaluation metrics used offer useful insights into the models' strengths and limitations, directing further study. To further improve the capabilities of conversational agents in task-oriented dialogue systems, it is imperative to address the outlined issues and consider new avenues. By doing this, we can open the door for numerous practical applications that include human-computer interactions that are more effective and efficient.

Bibliography

[1] Yohan Lee. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, 2021.

[2] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, 2018.

[3] Yichi Zhang, Zhijian Ou, and Zhou Yu. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611, 2020.

[4] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[6] Christian Buck, Kenneth Heafield, and Bas Van Ooyen. N-gram counts and language models from the common crawl. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3579–3584, 2014.

[7] Trieu H Trinh and Quoc V Le. A simple method for commonsense

reasoning. *arXiv preprint arXiv:1806.02847*, 2018.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[10] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

[11] M Guo, J Ainslie, D Uthus, S Ontanon, J Ni, YH Sung, and Y Yang. Longt5: Efficient text-to-text transformer for long sequences (2021). *URL https://arxiv. org/abs/2112.07916.*

[12] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[13] Joshua Ainslie, Santiago Ontañón, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. Etc: Encoding long and structured data in transformers. 2020.

[14] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

[15] Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

[16] Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jian-

guo Zhang, and Jindong Chen. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. *arXiv preprint arXiv:2007.12720*, 2020.

[17] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[18] Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. Mintl: Minimalist transfer learning for task-oriented dialogue systems. *arXiv preprint arXiv:2009.12005*, 2020.

[19] Tiancheng Zhao and Maxine Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*, 2016.

[20] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

[21] Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*, 2019.

[22] Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019.

[23] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Modelling hierarchical structure between dialogue policy and natural language generator with option framework for task-oriented dialogue system. *arXiv preprint arXiv:2006.06814*, 2020.

[24] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191, 2020.

[25] Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 583–592, 2020.

[26] Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*, 3, 2020.

[27] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[28] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[29] Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*, 2018.

[30] Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv preprint arXiv:1805.11004*, 2018.

[31] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[32] Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] Xiaodong Liu, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888*, 2018.

[34] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

[35] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[36] John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

[37] Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*, 2018.