TÉCNICO LISBOA

Domain Adaptation for Neural Generative-Based Dialogue Systems

Rui Orlando Magalhães Ribeiro

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Alberto Abad Gareta
Prof. José David Águas Lopes

Examination Committee

Chairperson: Prof. David Manuel Martins de Matos
Supervisor: Prof. Alberto Abad Gareta
Member of the Committee:
Prof. Maria Luísa Torres Ribeiro Marques da Silva Coheur

January 2021

Agradecimentos

Quero agradecer a todos os que me acompanharam nestes cinco anos e me ajudaram a concretizar esta tese. Foram tempos muito importantes que sei que vou recordar para sempre.

Aos meus orientadores, que me apoiaram e guiaram durante este longo percurso, mesmo com as dificuldades que a situação atual causou.

Aos meus amigos, que me fizeram companhia e que me fizeram sentir em casa quando vim para Lisboa sozinho. Aos tempos que passámos juntos e nos divertimos e também ao apoio e motivação que me deram para concluir este curso.

À minha família e especialmente aos meus pais, que sempre me apoiaram e me deram tudo o que precisei para completar este curso, que tanto me felicitaram como me ”deram na cabeça” quando mais precisei. Obrigado do fundo do coração por todo o esforço que fizeram por mim, esta tese nunca seria escrita se não fossem vocês.

À Joana, que aturou os meus desabafos e problemas, que sempre me apoiou enquanto escrevia esta tese e que nunca se fartou de me ouvir falar dela. Obrigado pelo tempo que passámos juntos e por me fazeres sentir feliz.

Abstract

Current generative-based dialogue systems are data-hungry and fail to adapt to new unseen domains when only a small amount of target data is available. Additionally, in real-world applications, most domains are underrepresented, so there is a need to create a system capable of generalizing to these domains using minimal data. There has been some notorious effort to surpass the problem of data scarcity in machine learning, however, there have only been a few attempts to solve this problem in generative-based dialogue systems.

In this thesis, we analyze existing state-of-the-art approaches that aim to solve the problems mentioned above and propose a novel model to surpass previous models' limitations by combining transfer-learning with meta-learning. Our approach relies on the belief that in order to successfully generalize to new domains using minimal data, the model needs to: *1.* learn a general dialogue representation from a larger data source, and then fine-tune with few examples from the unseen domain; *2.* improve how the model learns by simulating low-resource fine-tuning in the source domains.

We evaluate both baselines and our model on the MultiWOZ dataset and report BLEU and Entity F1. Results show that our model achieves higher performance in terms of accuracy and data-efficiency when compared to previous state-of-the-art approaches.

Keywords

Dialogue systems; Domain adaptation; Transfer-learning; Meta-learning

Resumo

Os sistemas de diálogo atuais baseados em geração dependem demasiado da quantidade de dados e têm dificuldade em adaptarem-se a novos domínios quando apenas uma quantidade mínima de dados desse domínio está disponível. Além disso, no mundo real, a maioria dos domínios está sub-representada, portanto, existe a necessidade de criar um sistema que seja capaz de generalizar para estes domínios utilizando o mínimo de dados possível. Tem havido algum esforço para superar o problema da escassez de dados na inteligência artifical, no entanto, houve poucas tentativas em resolver esse problema em sistemas de diálogo baseados em geração.

Nesta tese analisamos as abordagens estado-da-arte existentes que visam resolver os problemas mencionados acima e propomos um modelo que tenta superar as limitações dos modelos anteriores, combinando aprendizagem por transferência com meta-aprendizagem. A nossa abordagem baseia-se na crença de que, para generalizar para novos domínios utilizando o mínimo de dados possível, o modelo precisa de: *1.* aprender uma representação geral do diálogo de uma grande fonte de dados e, em seguida, ajustar o modelo com apenas alguns exemplos do domínio sub-representado; *2.* melhorar a maneira como o modelo aprende, simulando pequenos ajustes na fase de treino.

Avaliamos as abordagens existentes e o nosso modelo no conjunto de dados MultiWOZ e relatamos as medidas BLEU e a Entity F1. Os resultados mostram que o nosso modelo alcança um desempenho superior em termos de precisão e de eficiência de dados quando comparado com as abordagens estado-da-arte anteriores.

Palavras Chave

Sistemas de diálogo; Adaptação de domínio; Aprendizagem por transferência; Meta-aprendizagem

Contents

List of Figures

List of Tables

Acronyms

ZSDG	Zero-Shot Dialogue Generation
FSDG	Few-Shot Dialogue Generation
DiKTNet	Dialogue Knowledge Transfer Network
DAML	Domain Adaptive Dialogue Generation via Meta Learning
VAE	Variational Auto-Encoder
DI-VAE	Discrete Information Variational Auto-Encoder
LAED	Latent Action Encoder-Decoder
DI-VST	Discrete Information Variational Skip-Thought
HRED	Hierarchical Recurrent Encoder-Decoder
SEQ2SEQ	Sequence-to-Sequence
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long-Short Term Memory Unit
GRU	Gated Recurrent Unit
MAML	Model-Agnostic Metal-Learning
KB	Knowledge-Base
DATML	Domain Adaptation using Transfer Meta-Learning
SOS	Start-of-Sequence
EOS	End-of-Sequence
NLU	Natural Language Understanding
NLG	Natural Language Generation
DM	Dialogue Manager

BLEU Bilingual Evaluation Understudy

ROUGUE Recall-Oriented Understudy for Gisting Evaluation

METEOR Metric for Evaluation of Translation with Explicit ORdering

1

Introduction

Contents

1.1 Problem and Motivation

With the appearance of chatbots like Siri and Alexa capable of having fluent and consistent conversations, dialogue systems have become very popular these days. Additionally, the emergence of deep learning techniques in natural language processing contributes to this popularity and various new models were created in order to surpass previous rule-based models. However, these generative-based models are data-hungry, they need large amounts of training data in order to obtain good results, they produce dull responses and fail to adapt to new unseen domains when only a few examples of data are available. Besides, in real-world applications, most of domains are underrepresented, so there is a need to create a model capable of generalizing to these domains using the minimum amount of data as possible.

Deep learning models pretrained on large-scale datasets have proven to generalize to unseen domains better than randomly initialized ones [5]. Following this idea, in order to successfully adapt to new domains, the system needs to learn the general dialogue structure and only has to obtain domain-specific knowledge from the unseen domain. For instance, using the analogy from [6], if a person is hired to work in the shoe department and after a few months is transferred to the clothing department, the worker only needs to learn some specific information about the new domain such as clothing sizes and types, and does not need to relearn how to converse with the customer. As this may seem simple in the human world, in the dialogue systems' setting this task is very difficult to succeed and even harder when only a few data is available.

1.2 Current Approaches and their Limitations

The reduced amount of available data has always been a problem in domain adaptation tasks. Methods as meta-learning [7], transfer learning [8–10] and few-shot learning [11–13] were introduced to solve this problem in machine learning. However, there were only a few attempts to solve the problem of domain adaptation in end-to-end dialogue systems.

Perhaps, the first study to pursue this direction was the work from Zero-Shot Dialogue Generation (ZSDG) [6], where the authors performed zero-shot dialogue generation using minimal data in the form of seed responses. The authors do not use complete dialogues and describe it as "zero-shot", however, the model still depends on human annotated data. As this approach seems promising, ZSDG relies on these annotations for seed responses, and in the real-world scenario, if collecting data for underrepresented domains is difficult enough, the access to annotated data becomes infeasible.

More recent studies attempt to perform domain adaption without the need of human annotated data and adopt the methods presented above: Domain Adaptive Dialogue Generation via Meta Learning (DAML) [14] incorporates meta-learning into the *sequicity* [15] model to train a dialogue system able to

generalize to unseen domains. This approach seems promising, yet DAML was evaluated on a synthetic dataset and should ideally be tested in real data for more realistic results. Another approach to solve the data-efficiency problem is Dialogue Knowledge Transfer Network (DiKTNet) [3], which applies transfer learning by leveraging general latent representations from a large data-source and incorporating them into a Hierarchical Recurrent Encoder-Decoder (HRED). We will describe this model in detail in the following sections as it represents a key feature for our solution.

1.3 Proposed Approach

In this work, we study the importance of generalizing to unseen domains using minimal data and aim to design a novel model to surpass this problem. We believe that for successful adaptation to new domains, two key features are essential for improving the overall performance of a dialogue system: better representation learning and better learning techniques. Following this belief, we are concerned with the exploration of a method able to learn a more general dialogue representation from a large data-source and able to incorporate this information into a dialogue system.

We follow this reasoning and introduce Domain Adaptation using Transfer Meta-Learning (DATML), a dialogue system that combines both transfer-learning with meta-learning for the purpose of adapting to unseen domains. Our model improves the approach from DiKTNet by enhancing its learning method while keeping the strong representation learning present in both ELMo [16] contextual embeddings and latent representations. For that, we divide the training method into three training stages: a pre-training phase where the latent variables are leveraged from a domain-agnostic dataset; instead of performing joint training as in original work, we divide this stage into source training with all data except dialogues from the unseen domain and fine-tune using only a few examples from the target domain. We incorporate meta-learning in source training as this method proved to be promising at capturing domain-agnostic dialogue representations [14]. However, instead of using Model-Agnostic Metal-Learning (MAML) [7] algorithm, we use a first-order optimization-based method, Reptile [17], which has shown to achieve similar or even better results than MAML while being more lightweight in terms of memory consumption [18].

We evaluate our model in the MultiWOZ dataset [19] and compare our approach with both ZSDG and DiKTNet. As the code for both baselines is openly available online, we adapt and evaluate their implementations on the MultiWOZ corpus. Our model outperforms both ZSDG and state-of-the-art DiKTNet when the same amount of data is available. Furthermore, DATML achieves superior performance with 3% of available target data in comparison to DiKTNet with 10%, which shows that DATML surpasses DiKTNet in terms of both performance and data-efficiency.

1.4 Structure of the Document

This document is organized as follows. In section 2.1 and 2.2, we present some background necessary to follow this work and in section 2.3, we give an overview of typical dialogue systems. In section 2.4, we present some relevant datasets used to train these models, and in section 2.5, we analyze typical measures used to evaluate dialogue systems. In chapter 3, we detail the architecture of the domain adaptive models described above and analyze their limitations. In chapter 4, we describe our solution and detail the meta-learning algorithm employed in our model and in chapter 5, we show how we evaluated and compared DAML with both baselines. In chapter 6, we briefly summarize what we discussed in this thesis and propose some future work.

2

Background

Contents

2.1 Machine Learning Foundations

2.1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [20] are a class of neural networks that receive an input $\mathbf{x} = (x_1, x_2, ..., x_T)$ and maintain an internal hidden state $\mathbf{h}$. Unlike Feed-forward Neural Networks [21], RNNs take into account historical information by preserving an internal state that depends on previous hidden states. This allows RNNs to process sequences of inputs. More formally, the RNN updates its recurrent hidden state $\mathbf{h}_t$, at each time step t, by:

$$\mathbf{h}_t = f\left(x_t, \mathbf{h}_{t-1}\right), \tag{2.1}$$

where f is a non-linear activation or gating function such as a simple logistic sigmoid function or more complex functions, e.g., Gated Recurrent Unit (GRU) [22] or its generalization, Long-Short Term Memory Unit (LSTM) [23]. These complex functions were introduced to overcome the vanishing/exploding gradient problem [24], where the multiplicative gradient can exponentially decrease/increase with respect to the number of layers and cause RNNs to fail at capturing long term dependencies. Optionally, the RNN may have an output $\mathbf{y}$ of variable size.

A generative RNN can describe a probability distribution over sequences of inputs by being trained to predict the next output from the sequence. Thus, the probability of a sequence of size T can be modeled as:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_{t-1}, ..., x_1), \tag{2.2}$$

and each conditional probability can be described as:

$$p(x_t | x_{t-1}, ..., x_1) = softmax(\mathbf{h}_t), \tag{2.3}$$

where $\mathbf{h}_t$ is from Eq. 2.1. If we consider $\mathbf{x}$ as being a sentence and x_n being the n'th word processed so far from that sentence, we can use a generative RNN to predict the next word x_{n+1} of that sentence (see Fig. 2.1).

Convolutional Neural Networks (CNNs) [25] are another popular class of neural networks and have achieved impressive results in dialogue act classification [26] and emotion recognition [27]. However, as there has been little work using CNNs at domain adaptation in dialogue systems and the models that we base our work on all use RNNs, we chose to describe RNNs in more detail.

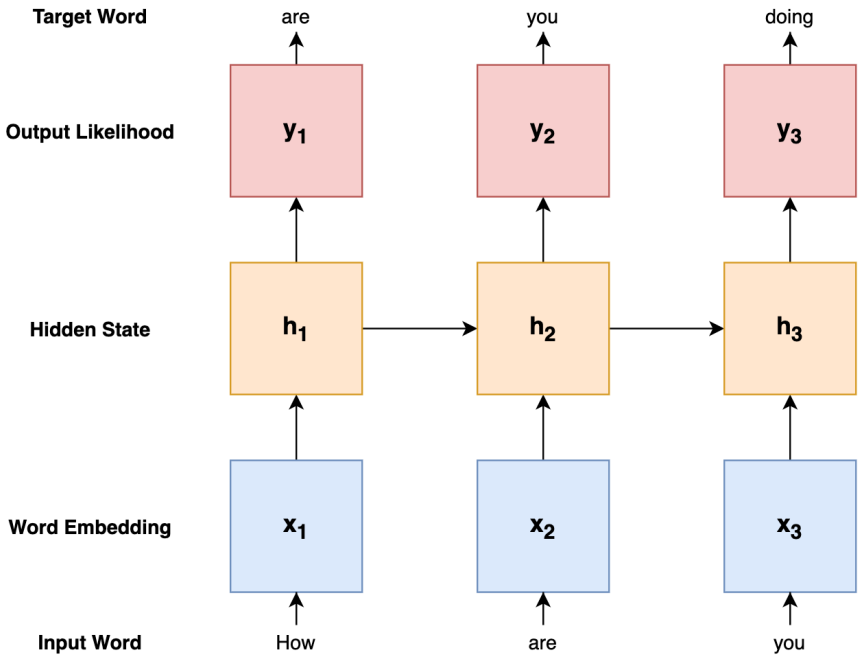

Figure 2.1: An example of a generative RNN predicting the next word from a sentence. Here, the RNN already processed *"How are you"* and predicts the next word *"doing"*. The input vector is the word representation and can be a one-hot vector or a word embedding, e.g., word2vec [1] or gloVe [2].

2.2 Encoder-Decoder Models

As presented above, neural networks are powerful machine learning models and have achieved remarkable results in domains such as acoustic modeling in speech recognition [28] and image classification [29, 30]. However, in settings such as dialogue generation and machine translation, the input and output from the model are better expressed as sequences of inputs and outputs, respectively.

2.2.1 Sequence-to-Sequence Model

To surpass this, [31] introduced the Sequence-to-Sequence (SEQ2SEQ) framework and proved that RNNs could be adapted to map complex structures to other structures instead of just resolve classification problems. These models enable to encode the whole sequence into one vector and to decode a vector representation back into a sequence. This architecture calculates the conditional probability of an output sequence $\mathbf{y} = (y_1, y_2, ..., y_L)$ given the input sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$, e.g $p(y_1, ..., y_L | x_1, ..., x_T)$, where the lengths of the sequences L and T may differ.

The encoder processes the input sequence word by word, updating the recurrent hidden state according to Eq. 2.1. After processing the whole sentence, the last hidden state contains all information about the input sequence, represented as the context $\mathbf{c}$.

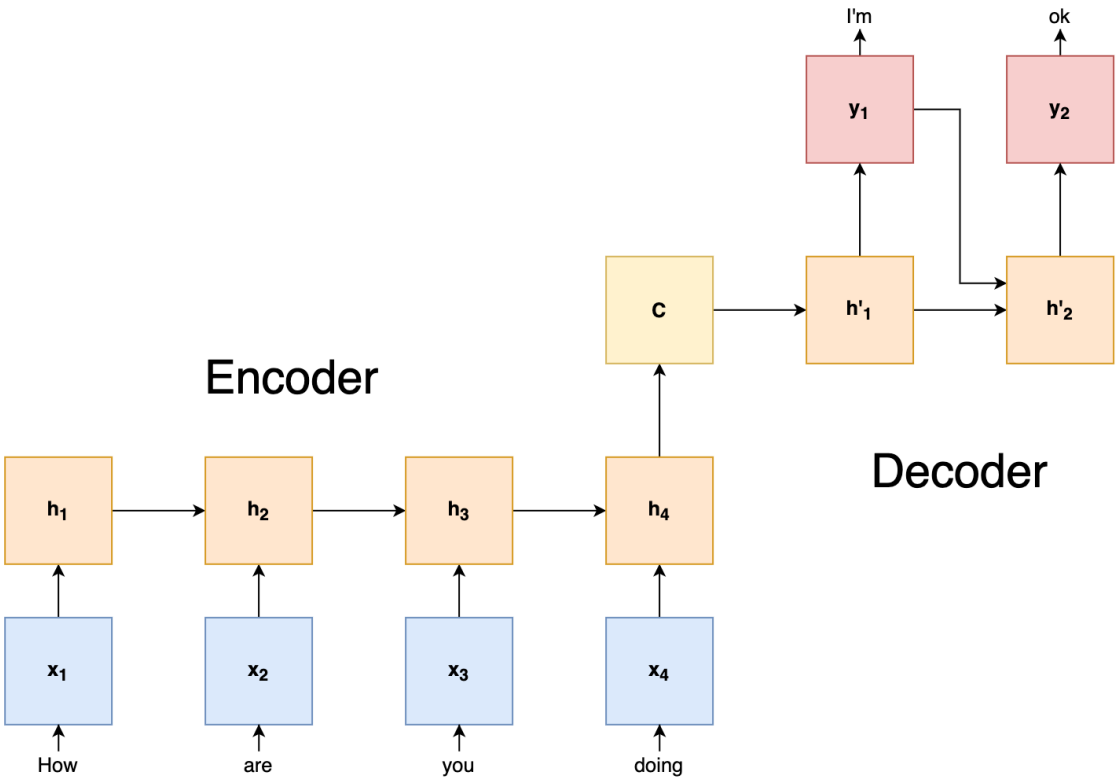

Figure 2.2: An application of the encoder-decoder model. Here, the encoder processes the sequence *"How are you doing"* and the decoder generates the output sequence *"I'm ok"*. The decoder takes as input the last hidden state from the encoder, the context c, and the last generated output.

Both encoder and decoder generate the output sequence given the last hidden state. However, in contrast with the encoder, the output word y_t and $\mathbf{h}_t$ from the decoder depend from the last generated symbol y_{t-1} and from the context $\mathbf{c}$ from the encoder. More formally, the hidden state from the decoder is updated by:

$$\mathbf{h}_t = f\left(y_{t-1}, \mathbf{h}_{t-1}, \mathbf{c}\right), \tag{2.4}$$

and the probability for the next generated word can be described as:

$$p(y_t|y_{t-1}, ..., y_1, \mathbf{c}) = softmax(y_{t-1}, \mathbf{h}_t). \tag{2.5}$$

In Fig. 2.2, we provide an application of the SEQ2SEQ model, where the framework encodes a question and generates an appropriate response. This framework is essential to our project as it is adopted in all our baseline models and also in our proposed solution.

2.2.2 Hierarchical Recurrent Encoder-Decoder

HRED [32, 33] generalizes the encoder-decoder framework to the dialogue environment. This improved framework aims to capture the word-level and utterance-level structure, as dialogue can be modeled as a sequence of utterances and the utterances as a sequence of word tokens. More formally, HRED model is a two-level hierarchy representation where the probability of dialogue $\mathbf{d}$ can be described as:

$$p(\mathbf{d}) = \prod_{n=1}^{N} p(\mathbf{d}_n|\mathbf{d}_{n-1}, ..., \mathbf{d}_1) = \prod_{n=1}^{N} \prod_{m=1}^{M_n} p(w_{n,m}|w_{n,<m}, \mathbf{d}_{<n}), \tag{2.6}$$

where $\mathbf{d}_n$ is the n'th utterance in the dialogue $\mathbf{d}$, $w_{n,m}$ is the m'th word of the n'th utterance, and M_n is the size of the n'th utterance.

In addition to the encoder RNN and the decoder RNN from the encoder-decoder model, HRED framework also includes a context RNN. In this model, each utterance is mapped by the encoder and the last hidden state of the encoder is the input of the context RNN, which updates its internal state with the complete utterance encoding and maintains all the information until that utterance. The decoder now takes as input the hidden state from the context RNN and acts as the module that generates the response.

As mentioned above, this framework was introduced to extend the encoder-decoder model to the dialogue setup and is important to our work for that reason. Additionally, it is also applied by some of our baseline systems and will be employed in our architecture as well.

2.3 Dialogue Systems Overview

In this section, we provide an overview of dialogue systems and review relevant applications of these models. We structure this analysis with the work from [34].

2.3.1 Frame-based Dialogue Systems

These dialogue systems, also known as pipeline-based dialogue systems, are one of the most important and successful models used in goal-oriented systems. These systems aim to solve tasks where the goal, for instance, is to book a flight: the agent needs to understand the user's intent and converse with him until it collects all the necessary information to successfully make the reservation. Typically, frame-based dialogue systems are divided into three main components as shown in Fig. 2.3:

1. **Natural Language Understanding (NLU).** In this module, the system maps utterances into pre-defined semantic frames. See table 2.1 for an example of a NLU representation.

2. **Dialogue Manager (DM).** This module is divided into two sub-modules: the dialogue state tracking that manages the input and outputs the current dialogue state, and the policy learning where the next action is decided.

3. **Natural Language Generation (NLG).** NLG generates the response given the action decided by the DM.

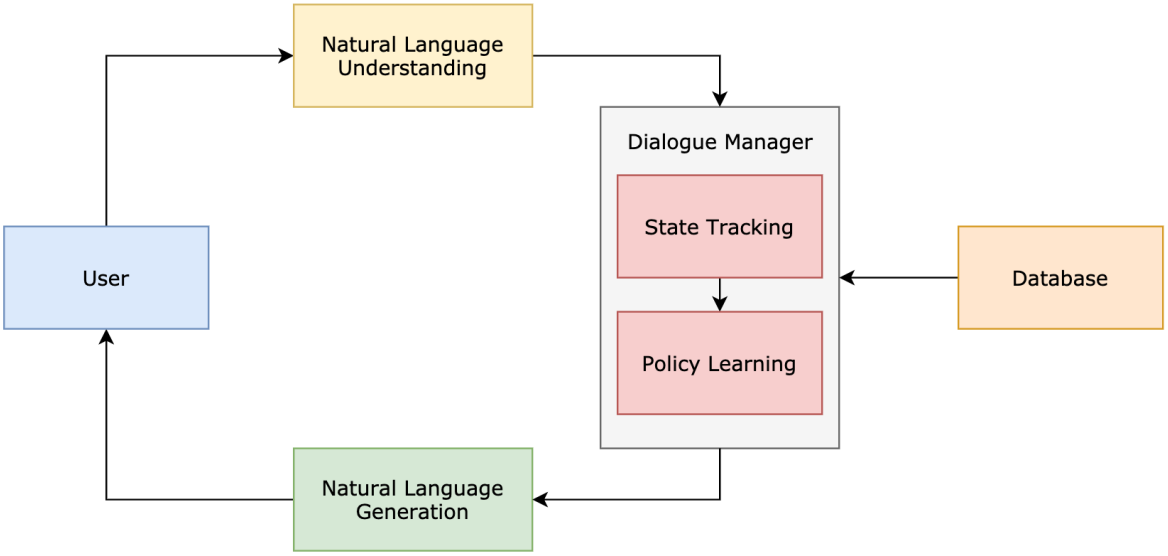

Figure 2.3: A typical pipeline for goal-oriented dialogue systems.

This dialogue system was first introduced in 1977 for GUS [35], a virtual agent designed to help customers making flight reservations. Despite performing well at booking flights, the system could not adapt to other types of reservations and could only deal with a particular set of questions.

When parsing dialogue utterances, it is necessary to represent the input sentences in a consistent and informative way. Typically, in the NLU module, the utterance is described using two representations, an utterance-level and a word-level semantic representation. The intent of the user and the utterance's domain are adopted to describe utterances, and their accurate retrieval is fundamental for the system to contextualize and guide the conversation. For word-level representation, methods as named entity recognition and slot-filing are used to describe utterances: named entity recognition locates and classifies named entities from unstructured text into predefined categories, such as person names, organizations, and locations; slot-filing seeks to tag words that carry meaning into predefined slot-values. In table 2.1, we present an example of a possible representation for a message where the user requests a taxi.

Table 2.1: An example of a NLU representation on the taxi domain.

Utterance	I	want	a	taxi	from	Rossio	to	Alameda
Slots	-	-	-	-	-	Departure	-	Destination
Intent	Request Taxi							
Domain	Taxi							

2.3.2 Retrieval-based Dialogue Systems

The idea in these dialogue systems is to select a response from a database containing dialogue contexts and responses using a ranking algorithm that retrieves the best candidate responses. The earliest studies have focused on single-turn response matching [36–38], which only use the current utterance from the user to retrieve the most adequate answer. In most cases, the candidate response and the context utterance are encoded as vectors and their matching score is calculated based on those two vectors, using some matching function such as bi-linear matching. As only the last utterance from the user is employed to retrieve the best match, these models do not consider the previous context, which may be a limitation in multi-turn conversation. To overtake this problem, recent studies have focused on multi-turn response matching [39], where all past utterances are taken into consideration when selecting the best response. This results in a more consistent and contextualized response. [40] applied deep learning models in retrieval-based systems by training a neural context encoder to rank the correct response with higher probability.

These systems have various limitations due to the requirement of having a large database in order to achieve good results. In most cases, as the database becomes bigger, the time needed to select the best response grows, slowing down the test speed. Another limitation is that as the system selects utterances from existing responses in the database, it doesn't possess the capability of creating innovative responses, becoming unsuitable to generalize to unseen domains.

2.3.3 Neural Generative-based Dialogue Systems

Typically, generative-based dialogue systems employ the SEQ2SEQ framework, described in section 2.2, that encodes the dialogue history and generates an appropriate response. These dialogue systems, unlike the systems described above, require little to no domain knowledge or hand-crafted rules to learn meaningful semantic representations. This characteristic provides flexibility to the system as it is capable of generating novel responses that are not included in the training data.

The encoder-decoder model has been applied in various domains: [41] leveraged a large amount of one-round conversation from a micro-blogging service and trained an encoder-decoder framework to generate grammatically correct and meaningful responses; [42] applied the encoder-decoder to generate captions for videos, where the input was sequences of frames and the output was sequences of

words; [43] addressed problems of text summarization, as modeling keywords and generating rare or unseen words in the training data, using the SEQ2SEQ framework.

However, current generative dialog models rely on large and complete dialogue datasets in order to achieve great performance. As collecting large amounts of training data becomes a difficulty, this method proves to be a limitation for underrepresented domains.

2.3.4 Hybrid Dialogue Systems

Both neural generative-based and retrieval-based dialogue systems are still far from achieving perfect results, so there has been some effort to combine both methods. The *AliMe Chat* system [44] integrates information retrieval and encoder-decoder models: given a query, the system will find the best candidate responses from both the retrieval-based and the generation-based systems, merge the selected responses from both systems and output the best response from the list.

2.4 Datasets

In this section, we present some relevant datasets used to train and evaluate task-oriented dialogue systems. Here we describe MultiWOZ and MetalWOZ datasets, which are used to evaluate both baselines and our model.

2.4.1 MetalWOZ

MetalWOZ [45] is a dataset specifically constructed for the task of generalizing to unseen domains and is designed to help developing meta-learning models.

This dataset contains about 37k task-oriented dialogues in 47 domains, such as schedules, apartment search, alarm setting, and banking. The data was collected in a Wizard-of-Oz fashion where a person acted like a robot/system and another acted as the user. The participants were instructed to converse until the user query was satisfied. In table 2.2, we present an excerpt of a dialogue between a user and a robot where the user's task was to *"Ask how to win at the text adventure Zork"* and the robot's task was to *"Tell the user that you are programmed to help them play games, not win them"*.

2.4.2 MultiWOZ

Multi-Domain Wizard-of-Oz dataset [19] is a large-scale multi-domain corpus containing human-to-human conversations with rich semantic labels (dialogue acts and domain-specific slot-values) from various domains and topics, and, like MetalWOZ, was collected in a Wizard-of-Oz fashion.

Table 2.2: An excert of a conversation between a user and a robot from MetalWOZ dataset.

System	Hello how may I help you?
User	I want to know how I can win playing Zork?
System	I am programmed to help you play games, not win them
User	What games can you help me with?
System	Any game.

Table 2.3: An example of a dialogue utterance and respective annotation from MultiWOZ dataset.

System	I have two restaurants. They are Pizza Hut Cherry Hinton and Restaurant Alimentum.
User	What type of food do each of them serve?
Domain	Restaurant
Slot-Value (Name)	Pizza Hut Cherry Hinton, Restaurant Alimentum

This dataset has more or less 10k dialogues, where the data is split into 1k dialogues for each validation and testing tasks. Around 70% of dialogues have more than 10 turns which shows how complex is the corpus. The average number of turns are 8.93 and 15.39 for single and multi-domain dialogues respectively with more or less 115k turns in total. This amount of available training data is favorable for generative-based systems as they tend to be data-hungry. MultiWOZ also comes with a large Knowledge-Base (KB) information for each domain/task, which approximates to real-world applications as goal-oriented systems have to access database information to generate an informative and appropriate response to the user.

As various task-oriented dialogue systems are not only specific to one domain and may share various domains, MultiWOZ proves to be suitable to these systems as it contains approximately 70% multi-domain dialogues, from 2 to 5 shared domains.

The original dataset had substantial amount of noise in the dialogue utterances and respective annotations. MultiWOZ 2.1 [46] improves MultiWOZ dataset by reducing noise in over 32% of state annotations across 40% of dialogue turns from the original dataset. MultiWOZ 2.2 [47] improved annotation errors and ontology issues present in it's previous versions.

2.4.3 SimDial Data

SimDial dataset was introduced in [6] to evaluate the ZSDG framework and was also used to evaluate DAML and compare it to ZSDG. SimDial is a synthetic dialogue generator that generates multi-domain dialogues in restaurant, movie, bus, weather, restaurant-style and restaurant-slot domains. This dataset simulates communication noise and is composed of long multi-turn dialogues, which challenges dialogue

systems' performance.

As this seems useful for both ZSDG and DAML setups taking into account that these models depend on annotations, it would be more interesting to use real data instead of a synthetic one in order to evaluate the models' ability to adapt to real-world applications, and that is why we choose to use MultiWOZ as our test dataset.

2.4.4 Stanford Multi-Domain

Stanford Multi-Domain dialogue dataset from [48] contains human-to-human dialogues from 3 different domains: navigation, weather and schedule. This dataset was used to evaluate ZSDG, Few-Shot Dialogue Generation (FSDG) and DiKTNet. In each dialogue the system has to accomplish a specific task and has associated some KB information to complete that specific task.

When generalizing to unseen domains, it is important to evaluate the system by training and testing in different domains. This dataset proves to be a challenge for domain transfer as the dialogue structure differs between domains. However, all dialogues are single domain and the dataset and average dialogue length is considerably smaller than MultiWOZ, which makes MultiWOZ a more realistic and challenging corpus. The dataset's statistics are presented in table 2.4.

Table 2.4: Stanford Multi-Domain dataset information [3].

Statistic \ Domain	Navigation	Weather	Schedule
Dialogues	800	797	828
Utterances	5248	4314	3170
Avg. dialogue length	6.56	5.41	3.83

2.5 Evaluation Metrics

In this section, we present the most relevant metrics used to evaluate models in natural language processing applications, such as machine-translation and dialogue systems.

2.5.1 Bilingual Evaluation Understudy

Bilingual Evaluation Understudy (BLEU) score [49] evaluates the quality of the generated responses by comparing how many words in the machine-generated response appeared in the real gold response. Scores are calculated for individual generated responses and then averaged over the whole corpus to

estimate the overall quality of the system. BLEU outputs a number between 0 and 1, where values near 0 represent considerable different responses and near 1 represent more similar responses.

2.5.2 Entity F1

Entity F1 is a measure used to calculate a test's accuracy by considering both precision and recall. Usually, in dialogue systems, this metric is used to evaluate the model's ability to generate relevant entities from the underlying KB.

2.5.3 Recall-Oriented Understudy for Gisting Evaluation

Recall-Oriented Understudy for Gisting Evaluation (ROUGUE) [50] metric also evaluates the quality of the generated responses as BLEU, however, ROUGUE compares how many words in the real response appeared in the machine-generated response.

2.5.4 Metric for Evaluation of Translation with Explicit Ordering

Metric for Evaluation of Translation with Explicit ORdering (METEOR) [51] is a metric introduced for machine translation that is also adopted to evaluate dialogue systems. It contains several features that are not found in other evaluation metrics, such as synonymy matching and stemming, along with the standard exact word matching present in BLEU and ROUGUE.

3

Related Work

Contents

3.1 Zero-Shot Dialogue Generation

ZSDG [6] discusses the importance of obtaining dialogue knowledge from domains, learning descriptions that are able to obtain domain-specific information and generalize to new domains. This proves to be an interesting approach because instead of describing domains as a set of example dialogues (that, in fact, share various domains), the domain descriptors allow focusing on the unique characteristics of a domain.

ZSDG is introduced in order to generalize to unseen situations using minimal data and is described as a learning problem where:

- The training data contains dialogue data from source domains and domain descriptions from both the source and target domains;
- The model is evaluated on the target domain in a "zero-shot" fashion (without using any full dialogues from the target domain and only these domain descriptions).

ZSDG assumes that every domain has its own domain description and the goal is to correlate the unseen target domain descriptions with the seen source descriptions. The model $\mathcal{F}$ should perform at least as well at the target domain as a model trained specifically to that target domain. The authors describe the problem as follows:

$$\begin{aligned} \textbf{Train Data: } & \{\mathbf{c}, \mathbf{x}, d\} \sim p_{source}(\mathbf{c}, \mathbf{x}, d) \\ & \{\phi(d)\}, d \in D \\ \textbf{Test Data: } & \{\mathbf{c}, \mathbf{x}, d\} \sim p_{target}(\mathbf{c}, \mathbf{x}, d) \\ \textbf{Goal: } & \mathcal{F} : C \times D \rightarrow X \end{aligned}$$

where $\mathbf{c}$ and $\mathbf{x}$ are the context and the next response, respectively, d is the domain and ϕ is the descriptor of the respective domain.

The configuration of domain descriptions is fundamental to achieve a strong performance in the target domain. The authors propose seed responses as a representation that can be applied to different domains. Seed responses assume that the model can discover similarities between responses from different domains and that behavior learned in source domains can be reused in the target domains. The domain description for domain d is now described as:

$$\{\mathbf{x}, \mathbf{a}, d\}_{seed}, \tag{3.1}$$

where $\mathbf{x}$ is a seed response and $\mathbf{a}$ is its annotation. Annotations allow inferring the relationship between responses from different domains. For instance, a domain description for the flight domain could be:

- $\mathbf{x}$: *The plain departs from Lisbon at 6pm.*
- $\mathbf{a}$: *[Inform, loc=Lisbon, leaveAt=18:00]*, where *Inform* is a general dialogue act and *Lisbon* and *18:00* are slot values.

As the number of seed responses is quite smaller compared to the number of potential responses from a domain, the seed responses should cover more utterances that are unique to domains. With this, ZSDG is assuming that different domains share the same dialogue structure and that the system only needs sentence-level information to adapt to new domains.

The architecture of the base model for the ZSDG is called the Action Matching Encoder-Decoder and is essentially a HRED with an attention-based Pointer-Sentinel Mixture copying mechanism [52] $\mathcal{F}$ that aims to learn a cross-domain representation for all source domains and embody the knowledge from the domain descriptions to generate novel responses in the target domain. With this, the model receives two types of data in the training phase: *1.* dialogue batches containing available dialogue from the source domains in the form of $\{\mathbf{c}, \mathbf{x}, d\}$; *2.* domain descriptions from either source and target domains described in equation 3.1.

For the first type of data, the parameters are updated by minimizing the following loss function:

$$\mathcal{L}_{dialog}(\mathcal{F}, \mathcal{R}) = -\log p_{\mathcal{F}^d}(\mathbf{x}|\mathcal{F}^e(\mathbf{c}, d)) + \lambda \mathcal{D}(\mathcal{R}(\mathbf{x}, d) \parallel \mathcal{F}^e(\mathbf{c}, d)), \tag{3.2}$$

where $\mathcal{F}^e$ and $\mathcal{F}^d$ are, respectively, the encoder and decoder of the model, $\mathcal{R}$ is the recognition encoder that shares the latent space from both utterances and domain descriptions of all domains, λ is a constant hyper parameter and $\mathcal{D}$ is a distance function that calculates the distance of the two vectors.

For the second type of data, the parameters of $\mathcal{F}^d$ and $\mathcal{R}$ are updated by minimizing the following loss function:

$$\mathcal{L}_{dd}(\mathcal{F}^d, \mathcal{R}) = -\log p_{\mathcal{F}^d}(\mathbf{x}|\mathcal{R}(\mathbf{a}, d)) + \lambda \mathcal{D}(\mathcal{R}(\mathbf{x}, d) \parallel \mathcal{R}(\mathbf{a}, d)). \tag{3.3}$$

By optimizing both loss functions, the model enforces *1.* dialogue utterances with similar annotations are closer in the latent space and *2.* responses are also closer to their contexts in the same latent space. Following this, the model is capable of generalizing to the target domain using only minimal data that is the domain description of the target domain.

This model achieved state-of-the-art in domain adaptation for end-to-end dialogue systems, however, it relies on annotated data to achieve better performance, and maintaining a consistent annotation for all source and target domains proves to be a difficult task. In underrepresented domains, besides not

existing any annotated data for those domains, typically the data is minimal and unorganized, so the effort to collect and annotate that data grows exponentially.

The following models present methods to overcome these limitations by not considering any annotated data and only using raw dialogue data.

3.2 Few-Shot Dialogue Generation

Based on the ZSDG framework presented above, [53] introduces the FSDG model to adapt to unseen domains. However, unlike ZSDG, FSDG uses unannotated data to train the model and acts in a few-shot setup, as it uses full in-domain dialogues instead of domain descriptions.

As discussed in chapter 1, a worker being transferred from the shoe department to the clothing department only needs to obtain domain-specific knowledge from the clothing domain to successfully serve the customers, as the general way of approaching a customer was already learned in the shoe department. With this, the model needs to learn a domain-agnostic representation of dialogue in order to generalize to unseen domains, by leveraging from a greater data source.

In order to achieve that, the authors consider the Latent Action Encoder-Decoder (LAED) framework [54]. LAED is, in essence, a Variational Auto-Encoder (VAE) representation method that allows discovering interpretable meaningful representations of utterances into discrete latent variables. LAED introduces a recognition network $\mathcal{R}$ that maps an utterance to a latent variable $\mathbf{z}$ and a generation network $\mathcal{G}$ that will be used to train $\mathbf{z}$'s representation. The goal is to represent the latent variable $\mathbf{z}$ independently of the context $\mathbf{c}$, so it can capture general dialogue semantics. LAED is a HRED framework and the authors have introduced two versions of the model: Discrete Information Variational Auto-Encoder (DI-VAE) and Discrete Information Variational Skip-Thought (DI-VST).

DI-VAE works as a typical VAE by reconstructing the input $\mathbf{x}$ and minimizing the error between the generated and the original data. The loss function that optimizes the DI-VAE model can be described as:

$$\mathcal{L}_{DI-VAE} = \mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p_{\mathcal{G}}(\mathbf{x}|\mathbf{z})] - KL(q(\mathbf{z}) \parallel p(\mathbf{z})), \tag{3.4}$$

where $p(\mathbf{z})$ and $q(\mathbf{z})$ are respectively the prior and posterior distributions of $\mathbf{z}$.

DI-VAE model aims to capture utterance representations by reconstructing each word of the utterance. However, it is also possible to capture the meaning by inferring from the surrounding context, as dialogue meaning is very context-dependent. With this, the authors propose another version, the DI-VST, which is inspired by the Skip-Thought representation [55]. DI-VST uses the same recognition network from DI-VAE to output the posterior distribution $q(\mathbf{z})$, however, two generators are now used to predict both previous $\mathbf{x}_p$ and following $\mathbf{x}_n$ utterances. The loss function that optimizes DI-VST can now

be described as:

$$\mathcal{L}_{DI-VST} = \mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p^n_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}) \log p^p_{\mathcal{G}}(\mathbf{x}_p|\mathbf{z})] - KL(q(\mathbf{z}) \parallel p(\mathbf{z})). \tag{3.5}$$

As already discussed in the beginning of this section, FSDG is essentially ZSDG but without the domain descriptions and instead these domain-agnostic latent variables. The training is performed using only dialogue batches, but now in the form of $\{\mathbf{c}, \mathbf{x}, \mathbf{k}, d\}$, where $\mathbf{k}$ is the KB information. First, the two LAED models are pre-trained in a domain-agnostic dataset in order to capture reusable dialogue representations. Then, when training with minimal data of the unseen domain, both LAED models are incorporated with the FSDG mentioned above, and the new encoding function can be described as:

$$\begin{aligned} \mathcal{F}^e(\mathbf{c}, \mathbf{k}, d)) = \mathcal{F}^e_{DI-VAE}(\mathbf{c}, \mathbf{k}, d)) \\ \oplus \mathcal{F}^e_{DI-VST}(\mathbf{c}, \mathbf{k}, d)) \\ \oplus \mathcal{F}^e_{FSDG}(\mathbf{c}, \mathbf{k}, d)), \end{aligned} \tag{3.6}$$

where the resulting $\mathcal{F}^e$ will be optimized according to loss function from Eq. 3.2.

The general idea of pre-training in a large and more domain-agnostic dataset to infer general dialogue semantics and then fine-tuning using minimal data from the unseen domain proves to be an interesting idea when adapting to unseen domains. However, in FSDG, the method to incorporate the pre-training information with the data from the target domains seems to be inefficient as the final encoder is a concatenation between the LAED and FSDG encoders, and the FSDG model is not conditioned on the representations learned from pre-training both the LAED models. Bellow, we present an approach by the same authors introduced to surpass these limitations.

3.3 Dialogue Knowledge Transfer Network

The basic idea from DiKTNet [3] is the same as FSDG: learning from a large dataset of source domains and fine-tuning using minimal data from the target domains. DiKTNet base model follows the approach from ZSDG, an HRED with an attention-based copying mechanism.

KB information is extremely important in task-oriented systems. For instance, if a user requests a restaurant near his location, the system has to query from a database the possible candidate restaurants and incorporate this information when answering to the user. Additionally, it is not expected for the model to recognize some of the KB information as it may contain unseen words, especially in the unseen domains. DiKTNet incorporates this information by concatenating it to the dialogue and using the copy mechanism mentioned above.

More formally, the base model's HRED $\mathcal{F}$ is optimized according to the following loss function:

$$\mathcal{L}_{HRED} = \log p_{\mathcal{F}^d}(\mathbf{x}_{sys}|\mathcal{F}^e(\mathbf{c}, \mathbf{x}_{usr})), \tag{3.7}$$

where $\mathbf{x}_{usr}$ is the user's request and $\mathbf{x}_{sys}$ is the system's respective response.

Although each domain has its specific dialogue structure, every domain still shares a general representation. Thus, DiKTNet learns this domain-agnostic representation from a large data-source and uses the same LAED models from FSDG to perform that task. However, now DiKTNet uses the DI-VAE model to obtain a latent representation of the user's request $\mathbf{z}_{usr}$ = DI-VAE($\mathbf{x}_{usr}$). As for the system's response, the model also wants to predict a latent representation $\mathbf{z}_{sys}$. In order to achieve that, DiKTNet uses the DI-VST model together with a context-aware hierarchical encoder-decoder that takes as input the user's request $\mathbf{x}_{usr}$ and the context $\mathbf{c}$. This encoder-decoder is different from the DI-VST for the reason that this new model, instead of predicting the previous and the following utterances, is interested in only predicting the following utterance that, in fact, is the system's response. The authors argue that DI-VAE captures the user utterance representation and that DI-VST predicts the system's action. When training with minimal data from the target domain, and after learning the latent representations $\mathbf{z}_{usr}$ and $\mathbf{z}_{sys}$, these variables are incorporated into the HRED $\mathcal{F}$ by an updated version of the loss function from equation 3.7:

$$\mathcal{L}_{HRED} = \mathbb{E}_{p(\mathbf{x}_{usr},c)p(\mathbf{z}_{usr},\mathbf{x}_{usr})p(\mathbf{z}_{sys}|\mathbf{x}_{usr},c)}[\log p_{\mathcal{F}^d}(\mathbf{x}_{sys}|\{\mathcal{F}^e(\mathbf{c}, \mathbf{x}_{usr}), \mathbf{z}_{usr}, \mathbf{z}_{sys}\})], \tag{3.8}$$

where $\{\ \}$ is the concatenation operator. With this, the authors ensure that the decoder is conditioned on the latent representations inferred in the pre-training phase and can now fine-tune in the target domain by taking into account that domain-agnostic representations. DiKTNet is also augmented with ELMo's [16] deep contextualized representations as word embeddings.

This model surpasses previous state-of-the-art results from ZSDG and will be the base model of our work. Bellow, we will present another promising method to overcome the problem of generalizing to underrepresented domains.

3.4 Domain Adaptive Dialogue Generation via Meta Learning

As mentioned in chapter 1, there has been a lot of effort to solve the problem of data scarcity in machine learning and new meta-learning algorithms have emerged. One of them was the MAML algorithm [7], which was introduced for few-shot learning. This method was designed to quickly adapt to new tasks using only a few training examples. To achieve this, MAML builds an internal representation across multiple domains by focusing on learning common representations instead of the distinctive features of

each domain.

DAML [14] incorporates this algorithm into the *sequicity* model [4], as MAML can be adopted in any gradient descent based model. The *sequicity* is a variation of the SEQ2SEQ framework with the addition of belief spans, which are text spans that track the belief at each turn. Essentially, the *sequicity* model decodes a belief span to facilitate KB search and then decodes a system response depending on the result of the query and the belief span (see table 3.1).

Table 3.1: An example of a dialogue where the user requests information on the restaurant domain. In essence, belief spans save the belief state at each turn with informative slot-values [4].

User:	Can I have some Italian food please?
Belief Span:	<Inf>Italian</Inf><Req></Req>
System:	What price range are you looking for?
User:	I want cheap ones.
Belief Span:	<Inf>Italian;cheap</Inf><Req></Req>
System:	Jamie's Italian is a cheap restaurant serving western food.

More formally, the *sequicity* model $\mathcal{M}$ can be described using the following equations:

$$
\begin{aligned}
h_t &= \text{Encoder}(B_{t-1}, R_{t-1}, U_t) \\
B_t &= \text{BeliefSpanDecoder}(h_t) \\
R_t &= \text{ResponeDecoder}(h, B_t, m_t),
\end{aligned}
$$

where B_t is the belief span at time step t, h_t is the hidden state at time step t and R_t is the generated response at time step t. m_t is a label that checks the availability of the information in the database, and can take the values *"no match"*, *"exact match"* and *"multiple matches"*.

The authors incorporate MAML into the *sequicity* model by changing the gradient update at the training phase. In MAML, there are two gradient update steps instead of one: (1) First, the model $\mathcal{M}$ is combined with the training data from each source domain separately. Then, (2) for each domain, the loss $\mathcal{L}$ is calculated and (3) a new temporary model $\mathcal{M}'$ for each domain is updated with the respective loss. After that, (4) the training data and the temporary model $\mathcal{M}'$ from each domain are used to calculate a new loss $\mathcal{L}'$. Finally, (5) all the losses calculated from each domain are summed and (6) the resulting loss is used to update the original model $\mathcal{M}$. The purpose of this update method is that the loss calculated from the updated model allows inferring the common dialogue structure from each domain instead of describing the different representations between domains.

This approach achieves promising results when compared to ZSDG, however, it is dependent on

dialogue annotation to construct it's belief spans and, as we discussed in ZSDG, the effort to collect and annotate data may be a difficulty in underrepresented domains. Additionally, this model was only evaluated in a synthetic dataset and needs to be tested in real human-to-human corpora to prove it's efficiency.

3.5 Summary

In chapter 3, we described the most relevant approaches to the problem of domain adaptation in end-to-end dialogue systems. DiKTNet will be employed in our solution as the base model architecture. The code for all approaches was openly available online, and we make use of their implementation to evaluate ZSDG and DiKTNet on MultiWOZ corpus (see chapter 5 for more details). Tables 3.2 and 3.3 present a summarized comparison between these models in terms of base architecture, usage of annotated data and adopted datasets for both training and evaluation.

Table 3.2: Summarized comparison between the models described in the previous sections. Additionally to the augmentation with ELMo embeddings, FSDG and DiKTNet also differ in the way the latent representations are incorporated into their model (see sections 3.2 and 3.3 for more details).

Models	Needs annotation	Seed responses	Latent representations	Belief spans	ELMo embeddings
ZSDG [6]	✓	✓	✗	✗	✗
FSDG [53]	✗	✗	✓	✗	✗
DiKTNet [3]	✗	✗	✓	✗	✓
DAML [14]	✓	✗	✗	✓	✗

Table 3.3: Continuation of table 3.2

Models	HRED	*Sequicity*	Uses MetalWOZ	Evaluates on Stanford Multi-Domain	Evaluates on SimDial
ZSDG [6]	✓	✗	✗	✓	✓
FSDG [53]	✓	✗	✓	✓	✗
DiKTNet [3]	✓	✗	✓	✓	✗
DAML [14]	✗	✓	✗	✗	✓

In the following section, we present our solution to the problem of generalizing to unseen domains using minimal data and demonstrate how we improved the DiKTNet model by updating its learning method.

4

Solution

Contents

4.1 Adapting MultiWOZ dataset

In order to compare the approaches described in chapter 3, we choose to use MultiWOZ dataset as our testing dataset. As presented in section 2.4, this corpus contains human-annotated information for each turn that is relevant for the models we want to compare: span annotations and user intent for each turn. It also keeps a large and complete KB and multi-domain dialogues, which make the task more realistic as task-oriented dialogue systems are not restricted to one domain and may share knowledge from distinct domains. However, some adjustments are needed in order to successfully evaluate both baselines and our model. In the following sections, we describe how we adapted the MultiWOZ for our setting.

4.1.1 Merging MultiWOZ 2.1 with 2.2

MultiWOZ 2.2 [47] was introduced to improve and identify annotation errors, inconsistencies and ontology issues present in its previous version. Additionally, slot span annotations were introduced for user and system utterances and the user intent was annotated for each turn. However, in order to follow the setting for ZSDG and DiKTNet, and to make the comparison as fair as possible, we need to preserve some representations that were abolished or simplified in the most recent version of MultiWOZ and are insufficient to build informative seed responses.

The authors improve the structure of the dataset's ontology by incorporating a new representation called schema that divides the different slots into *categorical* and *non-categorical* slots, which essentially are labelled according to the size of possible values: *categorical* are slots that can be fulfilled with a set of small finite values, typically less than fifty. For instance, the slot *hotel-internet*, which describes whether the hotel has internet, is categorical due to its possible values being *free*, *yes* and *no*; *non-categorical* have large or dynamic set of possible slot values and are retrieved from the dialogue history. *Restaurant-name* and *attraction-address* are examples of *non-categorical* slots, as they can take an enormous amount of possible values.

As this becomes an improvement in the overall structure of the dataset, this version lacks KB information and incorporates it directly into the dialogue history. Yet, all models presented above depend from a KB that serve as queries for each system utterance and so we keep that knowledge and merge it with MultiWOZ 2.2. The same applies for ontology and user intent: as in the newest version the intent is simplified and in order to be consistent with the authors, we keep the ontology from MultiWOZ 2.1.

4.1.2 Generating seed responses for ZSDG

As presented in chapter 3, ZSDG uses seed responses from both source and target domains in order to adapt to the new unseen domain. Here, experts annotated about 500 utterances for the 3 domains

present in Stanford Multi-Domain dataset. An example of a seed response used in their setting could be:

- **System**: The fastest route is 4 miles away with no traffic noted and sent to you navigation.
- **Seed response**: *inform #distance 4 miles #traffic no*

To be consistent with the author's original work, we also annotate 500 utterances using only the information present in MultiWOZ without the need for additional annotation. In order to achieve this, we use the dialogue acts present in MultiWOZ which describe the speakers's intent and slot values for each utterance. We only consider system utterances as user's dialogue acts in MultiWOZ are not very informative. We discard utterances without any intent or slot information. An example of a seed response in our setting could be:

- **System**: *We have 11 guest houses which are moderately priced, but no hotels.*
- **Seed response**: *hotel-inform #choice 11 #price moderately priced #type guest house*

4.1.3 Building KB for each dialogue

These models incorporate KB information into dialogue, simulating a database search for system's utterances. In Stanford Multi-Domain dataset, an average of 5 queries are provided for each dialogue. However, in MultiWOZ, a large KB is provided for the entire dataset, and as it becomes infeasible to incorporate the whole database into dialogues, we select a maximum of 7 KB entries for each dialogue.

In order to attain this, for each dialogue we go through each utterance and with string matching we retrieve the correct query with another possible candidates. For instance, if the user asks for "*cheap restaurants in the south*", we select all the queries present in the KB that match with the values *cheap* and *south*. When the system answers, for example, "*I have the Restaurant Alimentum that serves modern european food in the south*," we then select from the KB the ones that match with the values *restaurant alimentum* and *modern european food*, and append them to the previous queries. We do this for all utterances from dialogue and if we get more than 7 candidates, we randomly select 6 of them and maintain the correct KB query.

This information is useful in these models as they learn to identify the entities present in the dialogue and to choose which query satisfies the user's request.

4.2 Domain Adaptation using Transfer Meta-Learning

In this section, we propose our solution and describe how we merged transfer learning with meta-learning in order to generalize to unseen domains.

4.2.1 Base Model

Our base model architecture is the same as DiKTNet which, as presented in section 3.3, learns a more general representation that can be reused in any SEQ2SEQ model. For each turn, latent variables are generated using two different adaptations of the VAE model: DI-VAE, which learns meaningful latent representations based on the current response and DI-VST, that learns from the surrounding context. These learned variables are integrated into a HRED with an attention-based copying mechanism, by concatenating them with the last hidden states from the context encoder. The result is then combined into the decoder's initial state. We believe that representation learning is a key for domain adaptation, and that is why the combination of these latent variables with ELMo's [16] deep contextualized representations as word embeddings fits our purpose.

Instead of performing joint training as in original work, we first train the model with only source domains and then fine-tune it using a few example dialogues from the target domain. Below, we present how we enhanced our base model performance using an improved training strategy.

4.2.2 Meta-learning

As we referenced in chapter 1, better training techniques improve the overall system performance when adapting to new unseen domains using minimal data. In the following sections, we present our chosen meta-learning algorithm and describe how we adapted this algorithm into our base model.

4.2.2.A Model-Agnostic Meta-Learning

In section 3.4, we described DAML [14] which incorporates the MAML [7] algorithm into the *sequicity* model in end-to-end dialogue systems. This optimization-based meta-learning technique aims to learn a good initialization for the model on source domains that can be efficiently adapted to target domains using minimum fine-tuning.

More formally, in each iteration of MAML, two batches of the training corpus are sampled from a source domain d: $\mathcal{D}_s^d$ and $\mathcal{D}_q^d$ which are named, respectively, the *source* and the *query* set. Instead of calculating the gradient step and updating the model, in each episode low-resource fine-tuning is simulated: the model's parameters θ are preserved and for each domain d in source domains, new temporary parameters are calculated according to:

$$\theta^d = \theta - \beta \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_s^d), \tag{4.1}$$

where β is the inner learning rate. We could update the model's original parameters with the sum of the losses from all source domains, however, we choose to update the parameters after each domain iteration as this method performs better as presented by [56].

After each episode, the model's parameters are updated using the temporary ones calculated in equation 4.1:

$$\theta = \theta - \alpha \nabla_\theta \mathcal{L}(\theta^d, \mathcal{D}_q^d), \tag{4.2}$$

where α is the outer learning rate. As our model incorporates both context and KB information for each dialogue and as MAML also consumes to much memory, we instead adopt a lightweight version of the MAML algorithm that we describe below.

4.2.2.B Reptile

Reptile [17] algorithm is a first-order meta-learning algorithm where instead of sampling two source and query sets, $k > 1$ batches are retrieved for each domain $\mathcal{D}^d = (\mathcal{D}_1^d, ..., \mathcal{D}_k^d)$ and used to create the temporary model's parameters. The loss for the temporary model is calculated using Adam [57] optimizer according to:

$$\theta^d = \text{Adam}^k(\theta, \mathcal{D}^d, \beta), \tag{4.3}$$

where β is the inner learning rate and k is the number of updates in $\mathcal{D}^d$. After each episode, the model's original parameters are updated using the ones calculated in equation 4.3:

$$\theta = \theta + \alpha(\theta^d - \theta), \tag{4.4}$$

where α is the outer learning rate. Reptile is shown in [17] to produce equivalent or even better updates than MAML while consuming lower memory.

4.2.3 DATML

Our final model, DATML, is an adaptation of the architecture of DiKTNet with a better training technique, while maintaining the strong representation learning. In figure 4.1, we present a visual illustration of our approach. Instead of two training stages as in original work, we split joint training into source training and fine-tuning:

1. **Pre-training:** we maintain the first phase, where we learn the latent general representations for each turn using DI-VAE and DI-VST models. We exclude from training corpus all domains that may overlap with the unseen target domain.

2. **Source training:** in this phase, we exclude all data from the target domain and improve the training method by employing the Reptile meta-learning algorithm.

3. **Fine-tuning:** finally, we fine-tune the model using only few example dialogues from the target domain.

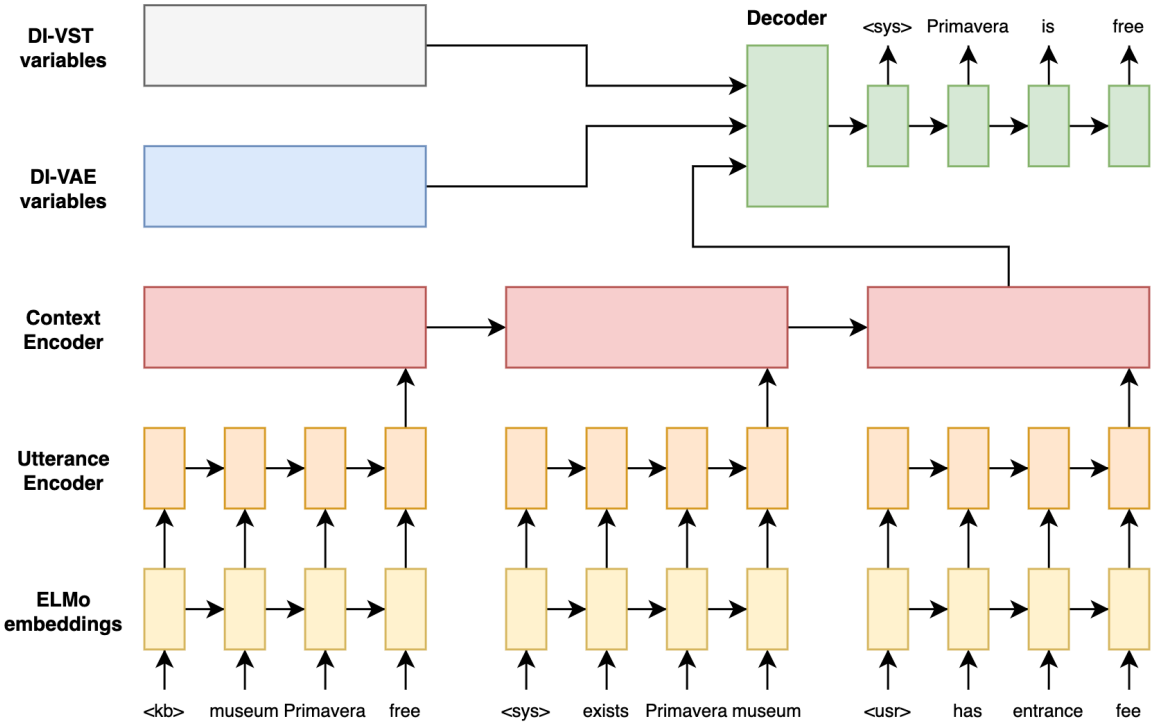

Figure 4.1: Visual illustration of both DiKTNet and DATML architectures. Both latent representations learned in the pre-training phase are concatenated with the last hidden state from the context encoder and become the initial hidden state of decoder. Start-of-Sequence (SOS) and End-of-Sequence (EOS) tokens are omitted for sake of simplicity.

In the following section, we demonstrate how we evaluate both baselines and our model and show that our DATML outperforms previous state-of-the-art DiKTNet and ZSDG approaches.

5

Evaluation

Contents

5.1 Experiments

In this section, we describe how we evaluated both ZSDG and DiKTNet baselines and DATML. We also analyze and suggest possible limitations of our approach.

5.1.1 Datasets

The dataset used to obtain the latent actions for DiKTNet and DATML was the MetalWOZ dataset. Both baselines and our approach were evaluated on MultiWOZ corpus. For more details about these corpora, see section 2.4.

5.1.2 Experimental Setup

In order to evaluate our model in a low-resource scenario, we choose the three most represented domains from MultiWOZ: hotel, restaurant and attraction, where each contains more than 1500 dialogues available. In the pre-training stage, we choose to learn the latent representations on MetalWOZ dataset as it is a domain-agnostic corpus introduced specifically for learning general representations. In order to make the evaluation as fair as possible, we exclude all dialogues from domains that could relate with the target domain. In table 5.1, we present the excluded domains from MetalWOZ dataset for each target domain from MultiWOZ.

Table 5.1: Excluded domains from MetalWOZ for each target domain in MultiWOZ dataset in pre-training stage.

Target domain from MultiWOZ	**Excluded domains from MetalWOZ**
hotel	*HOTEL_RESERVE*
restaurant	*MAKE_RESTAURANT_RESERVATIONS* *RESTAURANT_PICKER*
attraction	*EVENT_RESERVE*

We train both DI-VAE and DI-VST based LAED with y size of 10 and k size of 5, where y represents the number of latent variables and k the number of possible discrete values for each variable. Adam optimizer is used with a learning rate of 10^{-3} and Dropout $(p = 0.3)$ [58]. Both models' RNNs have hidden size of 512 and embedding size of 200 and were trained for 50 epochs, using early stopping if the validation accuracy does not improve on the the same number of already completed epochs.

For source training, we train DATML on MultiWOZ dataset and exclude all dialogues from the target domains. When fine-tuning to target domains, we use low resource data that varies from 1% to 10% by following [3] approach. We use the same hidden size for the model's encoder and decoder as in

pre-training phase. We also use Dropout $(p = 0.3)$. For Reptile, we use a k size of 5 and train the model for 4000 episodes. The inner and outer learning rates are 10^{-3} and 10^{-1}, respectively.

For ZSDG, we followed the original author's [6] setting and used 150 seed responses for each domain. These responses were generated as described in section 4.1. We use the same hidden size, learning rate and Dropout from DATML for both baseline models.

In order to fairly compare our model with state-of-the-art DiKTNet, we choose the same domain target data for both models by setting the random seed to 271, with no particular reason for selecting that number.

5.1.3 Metrics

We follow the work from DiKTNet [3] and ZSDG [6] and report BLEU and Entity F1 for each domain (see section 2.5, where these evaluation metrics are described in more detail).

5.2 Results and Discussion

Table 5.2 shows results on the three most-represented domains from MultiWOZ dataset. As observed in bold values, DATML outperforms both baselines ZSDG and DiKTNet in all low-resource scenarios.

Table 5.2: Results on the MultiWOZ dataset. We chose to evaluate the models on the three most represented domains.

Domain	**hotel**		**restaurant**		**attraction**	
Model	*BLEU %*	*Entity F1 %*	*BLEU %*	*Entity F1 %*	*BLEU %*	*Entity F1 %*
ZSDG	5.0	8.0	4.7	14.3	6.0	16.0
DiKTNet - 1%	10.7	17.3	12.4	17.5	10.2	18.6
DiKTNet - 3%	11.4	18.2	13.4	26.0	12.4	20.6
DiKTNet - 5%	11.6	17.6	16.6	25.7	12.0	27.1
DiKTNet - 10%	13.1	16.8	16.9	28.2	12.3	27.4
DATML - 1%	**10.9**	**18.0**	**14.1**	**24.0**	**11.0**	**23.4**
DATML - 3%	**13.0**	**23.1**	**16.7**	**28.4**	**14.1**	**28.6**
DATML - 5%	**14.1**	**25.3**	**17.8**	**30.0**	**15.0**	**31.2**
DATML - 10%	**14.2**	**26.3**	**18.3**	**32.9**	**15.4**	**32.2**

We investigate how the use of different amounts of target domain data impacts the system's performance. Table 5.2 shows that our model's performance correlates with the amount of available data from the unseen domain. While small improvements can be observed when only 1% of target domain data is

available, DATML achieves better results with 3% of target data in all metrics and domains in comparison to DiKTNet with 10% of available target data. This shows that DATML outperforms DiKTNet in terms of both performance and data-efficiency.

Table 5.2 also confirms that DiKTNet and DATML outperform ZSDG while using no annotated data and thus discarding human effort in annotating dialogues. This confirms that DATML achieves state-of-the-art results in data-efficiency and that is most suitable for real-world applications, as in underrepresented domains the amount of annotated data is almost nonexistent.

Tables 5.3, 5.4 and 5.5 show generated responses on the three domains by all the evaluated models above. In table 5.3, ZSDG identifies the domain in question, however, fails to provide a reference number for the user. DiKTNet shows a more similar response with the gold one, yet, also fails to provide a reference number. DATML succeeds in both reference selection and identifying the domain. In table 5.4, ZSDG identifies the domain with success but fails at constructing a coherent sentence. Although both models present similar structure in generated responses, DATML sucessfully identifies *Yu Garden* restaurant as the correct entity in comparison to DiKTNet's *Holiday Inn Cambridge*, which in the KB information represents a hotel and not a restaurant. Finally, in table 5.5, ZSDG fails to identify any entity from the KB, and DiKTNet fails to identify the domain in question. Although DATML identifies the domain and answers with a contextualized response, the sentence is not grammatically correct ("*to*" instead of "*for*") and it does not appear to be written by a human.

The results demonstrate that using optimization-based meta-learning improves the overall model's performance, and validate our initial idea that better learning techniques are a key feature when adapting to unseen domains using minimal data. Although the generated responses seem promising and DATML outperformed previous state-of-the-art DiKTNet, these low scores are far from being sufficient for real-world applications, and more work is essential to surpass the problem of data scarcity in dialogue systems.

Table 5.3: Examples of generated responses on MultiWOZ dataset for the hotel domain.

Context	Gold Response	ZSDG	DiKTNet - 1%	DATML - 1%
That sounds perfect. Can you reserve that and provide me with the reference number?	Okay, you're all set. The hotel is booked and the reference number is <ref>. Can I help you with anything else today?	I have booked your room. The reference number is hotel.	Sure, the reference number is down. Is there anything else I can help you with today?	Sure! Your booking was successful and your reference number is <ref>. Is there anything else I can help you with?

Table 5.4: Examples of generated responses on MultiWOZ dataset for the restaurant domain.

Context	Gold Response	ZSDG	DiKTNet - 1%	DATML - 1%
I am also looking for a restaurant that serves Chinese and that is in the same area as the hotel.	Yu Garden is an expensive restaurant in the same area as the hotel, would you like me to make reservations?	I have many 2 options for you . What type of cuisine would you like to be in?	How about the Express by Holiday Inn Cambridge?	How about the Yu Garden, which is in the east part of town?

Table 5.5: Examples of generated responses on MultiWOZ dataset for the attraction domain.

Context	Gold Response	ZSDG	DiKTNet - 1%	DATML - 1%
Hi, I'm looking for some places to go in Cambridge.	Sure, do you know what you would like to do?	I'm sorry, I don't have any matches. Would you like to look for something else?	In what area would you like to dine in?	What type of attraction are you looking to?

6

Conclusion

Contents

6.1 Conclusions

Domain adaptation in dialogue systems is extremely important as most domains are underrepresented in the real-world scenario. In this work, we leveraged state-of-the-art models that aim to surpass this problem and generalize to unseen domains using minimal data. We also presented relevant datasets and metrics used to evaluate these systems.

We proposed a model that improved previous state-of-the-art method by enhancing the training method. However, the evaluation results indicate that our model is far from being suited for real-world applications and shows that this field requires more study.

We would like to refer that, in addition to the proposed solution, we also experimented to:

- Substitute the ELMo embeddings from DATML with BERT-based [59] embeddings. BERT models have shown to outperform ELMo-based models in various natural language processing tasks. Essentially, our implementation received the dialogues as input and generated word embeddings for each token by concatenating the last four hidden layers from BERT. However, BERT's vocabulary size is considerable small for our setting and we would have to retrain the model with all new tokens.
- Started fine-tuning a GPT-2 transformer model [60] to our setting, but we could not get conclusive results in time and so this experiment is left for future work.

We also like to refer that along with this thesis, we have made a submission [61] which is currently under review.

6.2 Future Work

Domain adaptive dialogue systems are far from being suited to the real-word scenario, where most domains are underrepresented. Thus, more work is essential to surpass this problem of data scarcity.

We would like to follow the idea of incorporating BERT model as embedding layer in our solution, as it has been shown to achieve promising results. We also would like to continue the work with GPT-2 and explore newest models like BART [62] and other tranformer-based [63] models.

Bibliography

[1] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[2] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://www.aclweb.org/anthology/D14-1162

[3] I. Shalyminov, S. Lee, A. Eshghi, and O. Lemon, "Data-efficient goal-oriented conversation with dialogue knowledge transfer networks," in *EMNLP/IJCNLP*, 2019.

[4] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. 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)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1437–1447. [Online]. Available: https://www.aclweb.org/anthology/P18-1133

[5] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, p. 625–660, Mar. 2010.

[6] T. Zhao and M. Eskenazi, "Zero-shot dialog generation with cross-domain latent actions," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/W18-5001

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[8] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2016.

[9] D. George, H. Shen, and E. A. Huerta, “Deep transfer learning: A new deep learning glitch classification method for advanced ligo,” *ArXiv*, vol. abs/1706.07446, 2017.

[10] Y. Yao and G. Doretto, “Boosting for transfer learning with multiple sources,” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1855–1862, 2010.

[11] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4077–4087. [Online]. Available: http://papers.nips.cc/paper/6996-prototypical-networks-for-few-shot-learning.pdf

[12] V. G. Satorras and J. Bruna, “Few-shot learning with graph neural networks,” *ArXiv*, vol. abs/1711.04043, 2017.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] K. Qian and Z. Yu, “Domain adaptive dialog generation via meta learning,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2639–2649. [Online]. Available: https://www.aclweb.org/anthology/P19-1253

[15] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. 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)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1437–1447. [Online]. Available: https://www.aclweb.org/anthology/P18-1133

[16] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proc. of NAACL*, 2018.

[17] A. Nichol, J. Achiam, and J. Schulman, “On first-order meta-learning algorithms,” *CoRR*, vol. abs/1803.02999, 2018. [Online]. Available: http://arxiv.org/abs/1803.02999

[18] Z.-Y. Dou, K. Yu, and A. Anastasopoulos, “Investigating meta-learning algorithms for low-resource natural language understanding tasks,” 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)*. Hong Kong, China: Association

for Computational Linguistics, Nov. 2019, pp. 1192–1197. [Online]. Available: https://www.aclweb.org/anthology/D19-1112

[19] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic, "Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," in *EMNLP*, 2018.

[20] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *ArXiv*, vol. abs/1808.03314, 2018.

[21] M. Sazli, "A brief review of feed-forward neural networks," *Communications, Faculty Of Science, University of Ankara*, vol. 50, pp. 11–17, 01 2006.

[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *ICML*, 2012.

[25] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv e-prints*, 11 2015.

[26] S. Kawano, K. Yoshino, Y. Suzuki, and S. Nakamura, "Dialogue act classification in reference interview using convolutional neural network with byte pair encoding," in *9th International Workshop on Spoken Dialogue System Technology*, L. F. D'Haro, R. E. Banchs, and H. Li, Eds. Singapore: Springer Singapore, 2019, pp. 17–25.

[27] D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, "Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation," in *EMNLP/IJCNLP*, 2019.

[28] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[30] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 122–129, 2018.

[31] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," *Advances in NIPS*, 2014.

[32] I. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *AAAI*, 2015.

[33] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. G. Simonsen, and J. Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," *CoRR*, vol. abs/1507.02221, 2015. [Online]. Available: http://arxiv.org/abs/1507.02221

[34] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *ArXiv*, vol. abs/1711.01731, 2017.

[35] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "Gus, a frame-driven dialog system," *Artificial intelligence*, vol. 8, no. 2, pp. 155–173, 1977.

[36] A. C. Graesser, S. Lu, G. T. Jackson, H. H. Mitchell, M. Ventura, A. Olney, and M. M. Louwerse, "Autotutor: A tutor with dialogue in natural language," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 2, pp. 180–192, May 2004. [Online]. Available: https://doi.org/10.3758/BF03195563

[37] H. Wang, Z. Lu, H. Li, and E. Chen, "A dataset for research on short-text conversation," *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 935–945, 01 2013.

[38] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 1367–1375. [Online]. Available: http://papers.nips.cc/paper/5019-a-deep-architecture-for-matching-short-texts.pdf

[39] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Prague, Czech Republic: Association for Computational Linguistics, Sep. 2015, pp. 285–294. [Online]. Available: https://www.aclweb.org/anthology/W15-4640

[40] M. Wang, Z. Lu, H. Li, and Q. Liu, "Syntax-based deep matching of short texts," in *IJCAI*, 2015.

[41] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *ACL*, 2015.

[42] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - video to text," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[43] R. Nallapati, B. Zhou, C. N. dos Santos, Çaglar Gülçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *CoNLL*, 2016.

[44] M. Qiu, F.-L. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu, "AliMe chat: A sequence to sequence and rerank based chatbot engine," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 498–503. [Online]. Available: https://www.aclweb.org/anthology/P17-2079

[45] "A dataset of multi-domain dialogs for the fast adaptation of conversation models," https://www.microsoft.com/en-us/research/project/metalwoz/.

[46] M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, and D. Z. Hakkani-Tür, "Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines," *ArXiv*, vol. abs/1907.01669, 2019.

[47] X. Zang, A. Rastogi, S. Sunkara, R. Gupta, J. Zhang, and J. Chen, "MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines," in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Online: Association for Computational Linguistics, Jul. 2020, pp. 109–117. [Online]. Available: https://www.aclweb.org/anthology/2020.nlp4convai-1.13

[48] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, "Key-value retrieval networks for task-oriented dialogue," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Saarbrücken, Germany: Association for Computational Linguistics, Aug. 2017, pp. 37–49. [Online]. Available: https://www.aclweb.org/anthology/W17-5506

[49] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://www.aclweb.org/anthology/P02-1040

[50] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[51] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor,

Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. [Online]. Available: https://www.aclweb.org/anthology/W05-0909

[52] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *ArXiv*, vol. abs/1609.07843, 2016.

[53] I. Shalyminov, S. Lee, A. Eshghi, and O. Lemon, "Few-shot dialogue generation without annotated data: A transfer learning approach," *ArXiv*, vol. abs/1908.05854, 2019.

[54] T. Zhao, K. Lee, and M. Eskénazi, "Unsupervised discrete sentence representation learning for interpretable neural dialog generation," *CoRR*, vol. abs/1804.08069, 2018. [Online]. Available: http://arxiv.org/abs/1804.08069

[55] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *NIPS*, 2015.

[56] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HJGven05Y7

[57] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[59] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[60] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2018. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[61] R. Ribeiro, A. Abad, and J. Lopes, "Domain adaptation in dialogue systems using transfer and meta-learning," *Submitted to IberSPEECH 2020 Conference*.

[62] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, 2020. [Online]. Available: http://dx.doi.org/10.18653/v1/2020.acl-main.703

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.

A

Appendices

A.1 MetalWOZ full dialogue examples

Table A.1: MetalWOZ example dialogue from EVENT_RESERVE domain.

System	Hello how may I help you?
User	I need a tent for my son's birthday party.
System	Do you have any design in mind?
User	Circus tent.
System	I can make a reservation for that. Do you want to make one?
User	Yes
System	But it will be at an expensive cost. Do you still want to make a reservation?
User	I have the money. no worries.
System	It can be done. I'll reserve the tent for your son's bitrhday party.
User	How much is it?
System	$2000
User	That's so expensive.

Table A.2: MetalWOZ example dialogue from MAKE_RESTAURANT_RESERVATIONS domain.

System	Hello how may I help you?
User	Hi, I have a question.
System	I can help. What is your question?
User	Can you reserve me and my business partners a few tables at The New Yorker for tonight?
System	Unfortunately, there are no more tables available at The New Yorker.
User	How about at Cheesecake Factory?
System	Yes, there are tables available at the Cheesecake Factory. How many tables would you like to reserve?
User	I need three tables please.
System	For what time?
User	8 PM.
System	Okay, three tables have been reserved at the Cheesecake Factory tonight at 8 pm.

A.2 MultiWOZ full dialogue examples

Table A.3: MultiWOZ example dialogue from attraction and train domains.

User	Can you help me find a train leaving on tuesday departing from Cambridge?
System	And where would you like to go?
User	I would like to go to Kings Lynn, please.
System	At what time will you be traveling?
User	I would like to arrive by 11:45.
System	Train TR7548 leaves Cambridge at 10:11, arriving in Kings Lynn at 10:58. Would you like me to book your ticket?
User	Yes, please make a booking for 3 people.
System	Yes, the reservation is complete. You can pay 29.4 GBP at the station. The reference number is: GSD9FUA7. Can I help you further?
User	I am looking for museum type attractions in the centre of town.
System	I have found 11 museum attractions. Do you have a preference or would you like me to suggest one?
User	I would like one with free admission.
System	Primavera has free admission. Their address is 10 King's Parade.
User	Thank you very much, that is all I'll need.
System	Thank you very much.

Table A.4: MultiWOZ example dialogue from the hotel domain.

User	I am looking for a french restaurant.
System	There are two french restaurants listed. Do you have a preference of area? Centre or north part of town?
User	I don't care.
System	Cote is a French restaurant in the centre part of town, and it is in the expensive price range. Is that price range okay for you?
User	Are there any other French places available?
System	Restaurant Two Two is a nice place that serves French food in the expensive price range and is in the North part of town.
User	That sounds great! May I have their address and postcode, please?
System	Restaurant Two Two's Address is 22 Chesterton Road Chesterton, their postcode is C.B 4, 3 A.X. May I help you with anything else?
User	That's it. Thank you, goodbye
System	You're welcome. Goodbye.