Université de Montréal

Dialogue Systems Based on Pre-trained Language Models

par

Yan Zeng

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

Orientation Intelligence Artificielle

Juillet 2021

© Yan Zeng, 2021

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Dialogue Systems Based on Pre-trained Language Models

présenté par

Yan Zeng

a été évalué par un jury composé des personnes suivantes :

Stefan Monnier
(président-rapporteur)

Jian-Yun Nie
(directeur de recherche)

Guy Lapalme
(membre du jury)

Université de Montréal

Département d'informatique et de recherche opérationnelle

Ce mémoire intitulé

Dialogue Systems Based on Pre-trained Language Models

Présenté par

Yan Zeng

A été évalué par un jury composé des personnes suivantes

Stefan Monnier
Président-rapporteur

Jian-Yun Nie
Directeur de recherche

Guy Lapalme
Membre du jury

Résumé

Les modèles de langue pré-entraînés ont montré leur efficacité dans beaucoup de tâches de traitement de la langue naturelle. Ces modèles peuvent capter des régularités générales d'une langue à partir d'un grand ensemble de textes, qui sont utiles dans la plupart des applications en traitement de langue naturelle. Dans ce mémoire, nous étudions les problèmes de dialogue, i.e. générer une réponse à un énoncé de l'utilisateur. Nous exploitons les modèles de langue pré-entraînés pour traiter différents aspects des systèmes de dialogue.

Premièrement, les modèles de langue pré-entraînés sont entraînés and utilisés dans les systèmes de dialogue de différentes façons. Il n'est pas clair quelle façon est la plus appropriée. Pour le dialogue orienté-tâche, l'approche de l'état de l'art pour le suivi de l'état de dialogue (Dialogue State Tracking) utilise BERT comme encodeur et empile un autre réseau de neurones récurrent (RNN) sur les sorties de BERT comme décodeur. Dans ce cas, seul l'encodeur peut bénéficier des modèles de langue pré-entraînés. Dans la première partie de ce mémoire, nous proposons une méthode qui utilise un seul modèle BERT pour l'encodeur et le décodeur, permettant ainsi un ajustement de paramètres plus efficace. Notre méthode atteint une performance qui dépasse l'état de l'art.

Pour la tâche de génération de réponses dans un chatbot, nous comparons 4 approches communément utilisées. Elles sont basées sur des modèles pré-entraînés et utilisent des objectifs et des mécanismes d'attention différents. En nous appuyant sur des expérimentations, nous observons l'impact de deux types de disparité qui sont largement ignorées dans la littérature: disparité entre pré-entraînement et peaufinage, et disparité entre peaufinage et génération de réponse. Nous montrons que l'impact de ces disparités devient évident quand le volume de données d'entraînement est limité. Afin de remédier à ce problème, nous proposons deux méthodes qui réduisent les disparités, permettant d'améliorer la performance.

Deuxièmement, même si les méthodes basées sur des modèles pré-entraînés ont connu de grands succès en dialogue général, nous devons de plus en plus traiter le problème de dialogue conditionné, c'est-à-dire dialogue en relation à une certaine condition (qui peut désigner un personnage, un sujet, etc.). Des chercheurs se sont aussi intéressés aux systèmes de chatbot avec des habiletés de conversation multiples, i.e. chatbot capable de confronter différentes situations de dialogues conditionnés. Ainsi, dans la seconde partie de ce mémoire,

nous étudions le problème de génération de dialogue conditionné. D'abord, nous proposons une méthode générale qui exploite non seulement des données de dialogues conditionnées, mais aussi des données non-dialogues (textes) conditionnées. Ces dernières sont beaucoup plus faciles à acquérir en pratique. Ceci nous permet d'atténuer le problème de rareté de données. Ensuite, nous proposons des méthodes qui utilisent le concept d'adaptateur proposé récemment dans la littérature. Un adaptateur permet de renforcer un système de dialogue général en lui donnant une habileté spécifique. Nous montrons que les adaptateurs peuvent encoder des habiletés de dialogue conditionné de façon stricte ou flexible, tout en utilisant seulement 6% plus de paramètres.

Ce mémoire contient 4 travaux sur deux grands problèmes de dialogue: l'architecture inhérente du modèle de dialogue basé sur des modèles de langue pré-entraînés, et l'enrichissement d'un système de dialogue général pour avoir des habiletés spécifiques. Ces travaux non seulement nous permettent d'obtenir des performances dépassant de l'état de l'art, mais aussi soulignent l'importance de concevoir l'architecture du modèle pour bien correspondre à la tâche, plutôt que simplement augmenter le volume de données d'entraînement et la puissance de calcul brute.

Mots clés: Système de dialogue, génération de réponse conditionnée, Modèle de langue pré-entraîné, Apprentissage par transfert, peaufinage, adaptateur de dialogue

Abstract

Pre-trained language models (LMs) have shown to be effective in many NLP tasks. They can capture general language regularities from a large amount of texts, which are useful for most applications related to natural languages. In this thesis, we study the problems of dialogue, i.e. to generate a response to a user's utterance. We exploit pre-trained language models to deal with different aspects of dialogue systems.

First, pre-trained language models have been trained and used in different ways in dialogue systems and it is unclear what is the best way to use pre-trained language models in dialogue. For task-oriented dialogue systems, the state-of-the-art framework for Dialogue State Tracking (DST) uses BERT as the encoder and stacks an RNN upon BERT outputs as the decoder. Pre-trained language models are only leveraged for the encoder. In the first part of the thesis, we investigate methods using a single BERT model for both the encoder and the decoder, allowing for more effective parameter updating. Our method achieves new state-of-the-art performance.

For the task of response generation in generative chatbot systems, we further compare the 4 commonly used frameworks based on pre-trained LMs, which use different training objectives and attention mechanisms. Through extensive experiments, we observe the impact of two types of discrepancy: pretrain-finetune discrepancy and finetune-generation discrepancy (i.e. differences between pre-training and fine-tuning, and between fine-tuning and generation), which have not been paid attention to. We show that the impact of the discrepancies will surface when limited amount of training data is available. To alleviate the problem, we propose two methods to reduce discrepancies, yielding improved performance.

Second, even though pre-training based methods have shown excellent performance in general dialogue generation, we are more and more faced with the problem of conditioned conversation, i.e. conversation in relation with some condition (persona, topic, etc.). Researchers are also interested in multi-skill chatbot systems, namely equipping a chatbot with abilities to confront different conditioned generation tasks. Therefore, in the second part of the thesis, we investigate the problem of conditioned dialogue generation. First, we propose a general method that leverages not only conditioned dialogue data, but also conditioned non-dialogue text data, which are much easier to collect, in order to alleviate the data

scarcity issue of conditioned dialogue generation. Second, the concept of Adapter has been recently proposed, which adapts a general dialogue system to enhance some dialogue skill. We investigate the ways to learn a dialogue skill. We show that Adapter has enough capacity to model a dialogue skill for either loosely-conditioned or strictly-conditioned response generation, while using only 6% more parameters.

This thesis contains 4 pieces of work relating to the two general problems in dialogue systems: the inherent architecture for dialogue systems based on pre-trained LMs, and enhancement of a general dialogue system for some specific skills. The studies not only propose new approaches that outperform the current state of the art, but also stress the importance of carefully designing the model architecture to fit the task, instead of simply increasing the amount of training data and the raw computation power.

Keywords: dialogue system, conditioned response generation, pre-trained language model, transfer learning, fine-tuning, dialogue adapter

Contents

List of tables

List of figures

List of abbreviations

NLP	Natural Language Processing
NLU	Natural Language Understanding
NLG	Natural Language Generation
SMT	Statistical Machine Translation
Seq2Seq	Sequence to Sequence
RNN	Recurrent Neural Network
LSTM	Long Short-term Memory
GRU	Gated Recurrent Unit
GPT	Generative Pre-training
BERT	Bidirectional Encoder Representations from Transformers
LM	Language Model

RL	Reinforcement Learning
VAE	Variational Auto-encoder
MLE	Maximum Likelihood Estimation
NLLLoss	Negative Log Likelihood Loss
MLM	Masked Language Modeling
SOTA	State-Of-The-Art

Acknowledgments

I am extremely happy for my journey at University of Montreal as a graduate student. First of all, I am deeply indebted to my advisor Professor Jian-Yun Nie, who gave me the chance to study abroad and always supported me. In the last two years, Professor Nie has patiently provided me endless guidance on research, which shaped my skills of research and my ways of solving problems.

I would also like to express my deepest appreciation to the members of my thesis committee, Professor Stefan Monnier and Professor Guy Lapalme. Thank you for your support of my thesis work and giving me priceless suggestions.

My experience as a graduate student was wonderful. I have learned so much new knowledge about natural language processing, especially pre-training and fine-tuning, open-domain dialogue, task-oriented dialogue, language generation, conditioned generation, and graph neural networks. I started my journey for NLP and deep learning in 2017, and I had internship at the NLP department of Baidu for a year and a half, where I was introduced to the fascinating field of dialogue system. I would like to thank Zhi-Bin Liu and Dr. Hua Wu who had supported me and taught me a lot.

Last but not least, I would like to thank my family who brought me to this beautiful world and unconditionally loved and guided me. Special thanks to my dear mother, who is also my best friend. Thank you for always helping me think clearly and for giving me the courage to try.

Introduction

The rise of pre-trained language models

Deep learning methods have been successfully applied to many natural language processing (NLP) tasks. Many architectures have been proposed, such as recurrent neural networks (RNN), transformer, etc. A specific deep learning architecture can be used on a set of data to build a model for an NLP task. However, the training process is known to be time-, resource- and data-intensive: it usually requires a large amount of training data (texts) and high computation power due to the complexity of the model (the number of parameters involved). Many research work cannot afford to have the necessary data and resources to build a large neural model for the NLP task at hand. This naturally limits the capability of the resulting model.

Recently, a new paradigm, pre-training then fine-tuning, emerged in NLP research. It exploits a model that has been already trained on a large amount of texts in advance, resulting in what we call a pre-trained language model (LM). Then the pre-trained language model is adapted or fine-tuned for the specific NLP tasks. This new paradigm has shown to be very effective in various NLP tasks. Recently, several large pre-trained language models have been made publicly available by Google [1], OpenAI [2], etc. These pre-trained models are the results of training on a large amount of texts (e.g. Wikipedia, Google books) in an unsupervised manner, following different deep learning architectures. For example, GPT [**65**] is a large general language model capable of predicting the next words from the previous ones. It has been shown to benefit many generative tasks. BERT [3] [**15**] is another powerful model based on transformer and attention mechanism that helps in many natural language understanding tasks. The advantage of a large pre-trained language model stems from the large amount of data used in the training: Based on a large amount of texts, it is expected that a large pre-trained LM such as BERT can capture much of the regularities in a language such as word order or context dependencies. Such models can be used as the base models in many

[1]BERT: https://github.com/google-research/bert
[2]GPT: https://openai.com/blog/gpt-2-1-5b-release/
[3]BERT can mean a specific way to build a language model, or the model pre-trained on a large set of texts. In this thesis, we generally refer to the second meaning when we talk about pre-trained LM.

of the applications in natural language processing (NLP). The tremendous success of pre-trained LMs leads to the new paradigm of "pre-training then fine-tuning" in NLP research, including in both task-oriented and open-domain dialogue generation. In this paradigm, one takes a pre-trained language model (e.g. BERT) as the starting model, then uses a set of domain-dependent data to further fine-tune the model. The fine-tuning process aims to adapt the model (its parameters) to the specific task at hand (e.g. dialogue generation).

To show the impact of pre-trained LMs, we can notice that before the rise of pre-trained Transformer-based language models, studies in dialogue generation were all based on Recurrent Neural Networks (RNN) and Seq2Seq framework. Many methods have been specifically designed for the tasks and achieved good performance. For example, hierarchical RNN [**81**] for multi-turn response generation, CVAE [**110**] to generate diverse responses, CCM [**114**] leveraging knowledge graph for response, generation under different conditions, and so on. However, once pre-trained models are used, none of the above methods can compete with a vanilla pre-trained model that does not exploit extra knowledge. This is an obvious sign that pre-trained LMs can be powerful for the dialogue task. Furthermore, many previous ideas that worked well in RNN-based systems might not bring similar improvement to a Transformer-based approaches. For example, multi-turn response generation in a Transformer-based system does not apply a hierarchical model structure. Instead, it simply concatenates multi-turn dialogue history as the input of Transformer. This shows that the brute force of a pre-trained LM can be superior to some finely designed RNN models. The above observations make it clear that any state-of-the-art approach today should exploit pre-trained LMs. Therefore, this dissertation investigates the utilization of pre-trained language models for dialogue systems.

Hidden questions in pre-trained language models

Despite the success of pre-trained LMs in dialogue, we observe that people usually exploit their brute-force power without carefully analyzing if a utilization (an architecture) is appropriate for the task. In many cases, a simple utilization of a large pre-trained LM (e.g. GPT) can outperform many carefully designed methods that exploit less data. However, this does not mean that it is useless to care about the adequate design of model. Behind the success of pre-trained LMs, many questions remain unanswered.

The first set of questions are related to model architecture. For example, a pre-trained LM is trained in some way determined in the pre-training process. How could such a pre-trained model be used in encoder and decoder of a dialogue task? Would a pre-trained LM more appropriate than another for a task due to the inherent nature of the task? For example, some pre-trained LMs such as BERT are pre-trained using bidirectional attention,

while the task requires a unidirectional attention (left-to-right). Would this difference hinder the adequacy of the pre-trained LM for the task?

To address these questions, we study different architectures to use pre-trained LMs in different tasks. First, for task-oriented dialogue systems, recent works on Dialogue State Tracking (DST) [**107, 74, 37**] replace previous RNN encoder with BERT, and achieve better performance than RNN-only framework (RNN encoder and decoder) [**99**] by leveraging the rich general linguistic features encoded in BERT. However, the BERT model cannot be directly used for the decoder due to the fact that the decoder cannot use the same bidirectional attention as in BERT (i.e. when generating a word, one can only see the words generated before using left-to-right attention). Therefore, an RNN decoder is stacked upon BERT encoder for the generation step. This RNN is to be trained on the dialogue data from scratch, and it does not fully leverage the pre-trained LM. To fully exploit a pre-trained LM, we propose a framework consisting of a single BERT that works as both the encoder and the decoder, which has a flat encoder-decoder architecture allowing for more effective parameter updating. Our method achieves new state-of-the-art performance and can converge to its best performance much faster and in a more stable manner than the existing framework. This framework can also be extended to build a task-oriented dialogue system when applying the idea of "text-to-text" of Google T5 to incorporate the other two tasks of task-oriented dialogue.

Second, for generative chatbot systems (i.e. to generate responses of chit-chat), some researchers believe that fine-tuning GPT, a left-to-right language model, corresponds well to the dialogue generation task [**109, 50**], while some others [**18, 4**] show that fine-tuning BERT can also achieve state-of-the-art performance. It is unclear what pre-training architecture should be used for response generation. To figure out how to best exploit a pre-trained LM for dialogue generation, we compare 4 widely used frameworks on 3 public datasets, each in large and small scale. This reveals that all the 4 frameworks contain some discrepancy: pretrain-finetune discrepancy meaning that the LM is pre-trained in a way, but fine-tuned in a different way; and finetune-generation discrepancy, meaning that the model is fine-tuned in a way different from its utilization in generation. The extensive experiments on the datasets, especially on small-scale datasets, will show the impact of discrepancies. Therefore, some adjustments are required to make the process more adequate. To this end, we propose two methods to reduce discrepancies, both yielding improved performance.

The second set of questions are related to specific utilization of pre-trained LMs in specific dialogue situations. Even though pre-training based methods have shown excellent performance in general dialogue generation, we are more and more faced with the problem of *conditioned* conversation in order to control the style[**44**], topic[**100**], emotion[**113**], situation[**78**], knowledge[**17**] of the generated responses. Conditioned dialogue means that we have to tune the dialogue to meet the condition, i.e. to correspond to a dialogue style,

a topic, and so on. Researchers are also interested in multi-skill chatbot systems, namely equipping a chatbot with abilities to confront different conditioned generation tasks. Therefore, beyond the pre-trained Transformer-based frameworks for general dialogue systems, we will investigate how a general dialogue system can be tuned toward conditioned dialogue generation.

There are two categories of conditioned dialogue, i.e. loosely-conditioned response generation and strictly-conditioned response generation. For the former, a clear label designating the type of the response is required. For example, persona labels [**44**] designate the speaking styles of the responses, and topic labels [**100**] or emotion labels [**113**] specify topic-related or emotion-related vocabularies. For the latter, extra knowledge is generally required to determine the content of the response, such as a persona profile [**108**], a situation description [**70**], or a wikipedia paragraph [**17**]. Enhancing a general dialogue system with strictly-conditioned dialogue could be easy: a state-of-the-art strictly-conditioned method [**97**] can be easily added in other models as well [**86, 57**], by simply concatenating the extra knowledge with the dialogue history as the model input. However, in many practical situations, we have to deal with loosely-conditioned dialogue, where the system should dynamically determine the content of response given dialogue context and condition. Loosely-conditioned dialogue takes an important part in open-domain conversation [**111**]. An acute problem we encounter in loosely-conditioned dialogue is the scarcity of labeled responses, i.e. we can expect to have limited amount of dialogue data with conditions clearly labeled. In this dissertation, to alleviate the problem, we exploit labeled non-dialogue text data related to the condition, which are much easier to collect. These data can be, for example, texts written by the same person (for a persona condition), within the same topic domain (for a topic condition), etc. We propose a multi-task learning approach to leverage both labeled dialogue and text data. The 3 tasks jointly optimize the same pre-trained Transformer – conditioned dialogue generation task on the labeled dialogue data, conditioned language encoding task and conditioned language generation task on the labeled text data. Experimental results show that our approach outperforms the state-of-the-art models by leveraging the labeled texts, and it also obtains larger improvement in performance compared to the previous methods to leverage text data.

Pushing the conditioned dialogue further, one can expect that a general dialogue system can be adapted to have some specific skill. The Adapter approach aims to construct such conditioned dialogue system with light-weight adapters. In this dissertation, we study different ways to learn dialogue skills: 1) using auxiliary loss specifically designed for the skill; 2) using multi-task learning to learn the common part, i.e. skill, of several tasks; 3) using the concept of Adapter to decrease model capacity avoiding to learn diverse styles. The concept of adapter has been proposed in a recent study [**73, 33**]. We will show that Adapter has enough capacity to model a dialogue skill for either loosely-conditioned or strictly-conditioned generation. Meanwhile, an Adapter only uses 6% more parameters on top of a pre-trained

dialogue model. Thus, it is possible to build a multi-skill model by using a fixed base dialogue model, e.g. a large pre-trained model, and multiple small Adapters. Then, given a dialogue history, a system only needs to switch among skills using elaborate rules or a more explainable model. We believe that this framework could be a promising avenue for future generative dialogue systems.

In summary, this dissertation deals with two sets of questions that have not been extensively investigated in the literature relating to dialogue based on pre-trained LMs. The main contributions of our studies are as follows: (1) Our study sheds light on the hidden problems in the utilization of pre-trained LMs in dialogue systems. We show that it is important to consider the adequacy of an architecture for a task in order to avoid discrepancies. (2) In addition to general dialogue, we show that it is possible to generate dialogues for specific conditions. This paves the way for more purpose-oriented open-domain dialogue.

Thesis Structure

The rest of the dissertation is organized as follows:

- **Chapter 1: Foundational Work**
  This chapter gives an overview of the prior research that lays the foundation for this dissertation, including both work about dialog system and related methods in deep learning.
- **Chapter 2: Leveraging Pre-trained LM for Task-oriented Dialogue State Tracking**
  This chapter presents methods to leverage pre-trained language model for dialogue state tracking, a core component in task-oriented dialogue systems. We propose a novel method to use a single BERT to work as both the encoder and the decoder.
- **Chapter 3: Leveraging Pre-trained LM for Generative Chatbots**
  This chapter compares the 4 widely used frameworks that utilize pre-trained language models for open-domain dialogue generation on 3 public datasets each in large and small scale, and we analyze each framework based on the experimental results. Through extensive experiments, we observe pretrain-finetune discrepancy and finetune-generation discrepancy of each framework. Then, we propose two methods to reduce discrepancies that lead to improved performance.
- **Chapter 4: Multi-Task Learning based on Pre-trained LM for Conditioned Dialogue Generation**
  In the following two chapters, we focus on conditioned dialogue for generative chatbot systems. Specifically, this chapter proposes a simple and efficient multi-task learning approach based on pre-trained Transformer that leverages different labeled data, i.e. , dialogue and text, for conditioned response generation.

- **Chapter 5: Adapter based on Pre-trained Model for Dialogue Skill Learning**

  This chapter shows that previous works in dialogue skill learning mainly learn untransferable dialogue styles instead of skills. We propose several ways to learn dialogue skills. Based on our approach, it is possible to build a multi-skill model by using a fixed base dialogue model and multiple small Adapters.

Except the introductory Chapter 1, the works described in the other chapters have been published/submitted to conferences, or published on ArXiv. The description will be many in the same format as a research paper. The references to the publications are as follows:

Jointly Optimizing State Operation Prediction and Value Generation for Dialogue State Tracking Yan Zeng, Jian-Yun Nie paper: `https://arxiv.org/abs/2010.14061` code: `https://github.com/zengyan-97/Transformer-DST` (Corresponding to Chapter 2)
An Investigation of Suitability of Pre-Trained Language Models for Dialogue Generation – Avoiding Discrepancies Yan Zeng, Jian-Yun Nie paper: `https://arxiv.org/abs/2010.12780` code: `https://github.com/zengyan-97/Transformer-MLM-DiffFree` note: accepted by ACL 2021 (findings). (Corresponding to Chapter 3)
A Simple and Efficient Multi-Task Learning Approach for Conditioned Dialogue Generation Yan Zeng, Jian-Yun Nie paper: `https://arxiv.org/abs/2010.11140` code: `https://github.com/zengyan-97/MultiT-C-Dialog` note: accepted by NAACL 2021 (oral). (Corresponding to Chapter 4)
Learning Transferable Dialogue Skills Yan Zeng, Jian-Yun Nie note: in double-blind period (Corresponding to Chapter 5)

Table 0.1. List of our publications.

Chapter 1

Foundational Work

This chapter presents an overview of the prior research that paves the foundation for this dissertation. We will first go over the background of dialogue systems. Then we will summarize deep learning techniques that the rest of this dissertation will build on.

1.1. Dialogue Systems

There are two ways to implement dialogue systems, namely retrieval-based and generation-based. The former uses a dialogue history as query to search in a database of dialogue histories and returns the best matched response as the answer. In contrast, the generation-based approach builds on templates or encoder-decoder networks to generate a new response to the given dialogue utterance.

Although in retrieval-based systems, the retrieved responses are human generated so that always grammatical and diverse, they are limited in the following aspects comparing to generation-based systems: (1) they cannot generate novel responses that are not in the database, leading to poor generalization given a limited database; (2) the query time may become larger as the database becomes bigger, slowing down the response speed at testing time.

In this dissertation, we focus on generation-based systems. With the help of pre-training, recent generation-based systems can generate grammatical and diverse responses similar to retrieval-based systems. Meanwhile, they are able to generate novel responses that are more suited to the dialogue contexts.

1.1.1. Task-oriented Dialogue Systems

There has been a long history of task-oriented (/goal-directed) dialogue systems. Task-oriented dialogue is designed to accomplish some specific task such as hotel booking. This type of systems often use a pipeline approach [**88**]. The pipeline requires natural language

understanding (NLU) for dialogue state tracking (also known as belief state tracking), dialogue management for deciding which actions to take based on those beliefs, and natural language generation (NLG) for generating responses.

Traditionally, each component of task-oriented dialogue systems is trained independently with different supervision. The NLU module is trained on domain and intent labels. The dialogue management module employs dialogue belief and dialogue act labels. The NLG module accesses templatized or natural responses. The modular dependencies of these components, however, can lead to error propagation when information is not provided to subsequent modules in the pipeline [**52**], i.e. when an error is made in a module, the subsequent modules will be unable to correct it.

Particularly, for the NLG module, previous systems [**72, 69**] rely on either canned responses or templates for generation. It is difficult to design the rules and templates. Besides, these methods have been shown to generate utterances that sound very unnatural in context [**10**]. An improvement is the use of data-driven methods, e.g. Statistical Machine Translation (SMT) in translating internal dialogue state into natural language [**38**].

Recent studies in deep learning, e.g. based on RNN [**96**] or Transformer [**32**], suggest a much simpler way to build a task-oriented system. For example, recent work based on pre-trained language model [**32**] has shown superior performance by adapting the idea of "text-to-text" approach [**66, 67, 36**] specifically for task-oriented dialogue. Specifically, they fine-tune a pre-trained language model by multi-task learning on the 3 subtasks of task-oriented dialogue. Then the response is generated end-to-end given an input text. Such an approach allows for a better integration of the three sub-tasks and avoids error propagation.

1.1.2. Open-domain Chatbots

"Chatterbots" [**94, 35**] as called in history attempt to engage users, typically by leading the topic of conversation. These systems usually limit interactions to a specific scenario (e.g. a Rogerian psychotherapist), and use a set of template rules for generating responses. It is difficult to extend such an approach to conversations in general domains.

Data-driven approaches are currently more used in chatbot. This is due to the fact that we have much more informal, public conversations on social media websites such as Reddit and Twitter, making it possible to train models for conversational tasks. For example, one can train a statistical machine translation model (SMT) [**76**] to generate a response from an input, or a RNN-based encoder-decoder model [**81**]. These modern chatbot systems focus on making chit-chat, open-ended, open-domain conversations with humans.

However, end-to-end generative neural networks [**89, 81, 110, 114**] still have weaknesses that prevent them from being generally useful: they often respond to open-ended input in

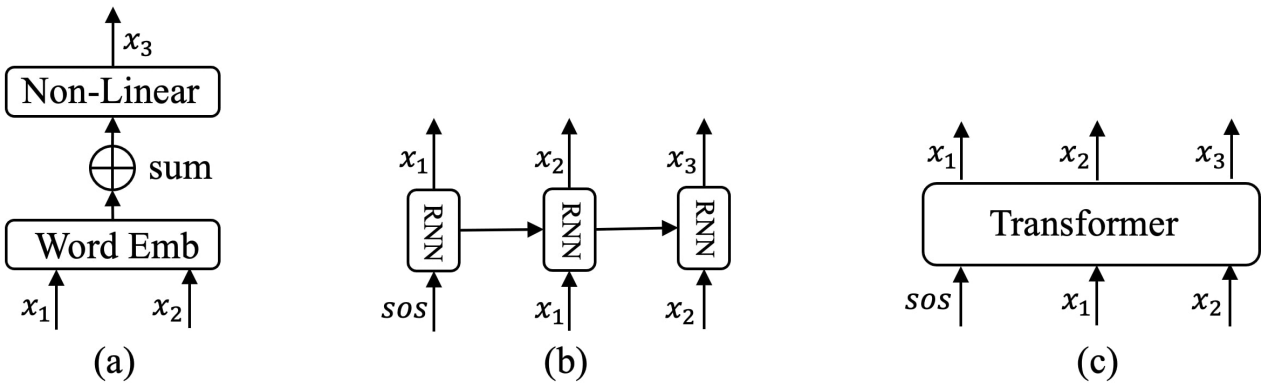

Fig. 1.1. (a) Neural network language model with 3rd Markov assumption; (b) RNN language model; (c) Transformer based language model. In this figure, these language models all apply auto-regressive training objective. "sos" is a special token representing "start of string". x_i is the i-th word of the text.

ways that do not make sense, or with replies that are vague and generic. Therefore, some open-domain chatbots such as MILABOT [**80**], XiaoIce [**115**], Gunrock [**12**] use hybrid, more complex frameworks, containing dialog managers coupled with knowledge-based, retrieval-based, rule-based, or generation-based systems.

Nevertheless, with progress in language pre-training on web text corpus and dialogue pre-training on large-scale data from social media, e.g. Twitter and Reddit, recent pre-training based methods [**109, 1, 77**] have shown excellent performance in generating fluent and diverse responses.

1.2. Deep Learning Foundations

1.2.1. Language Model (LM)

Architecture A statistical model of language can be represented by the conditional probability of the next word given all the previous ones,

$$P(w_t) = P(w_t|w_{t-1}, w_{t-2}, ..., w_1) \tag{1.2.1}$$

To train a language model, we maximize the likelihood of:

$$P(w_1, ..., w_T) = \prod_{t=1}^{T} P(w_t|w_{t-1}, w_{t-2}, ..., w_1) \tag{1.2.2}$$

Thus, we minimize the loss function:

$$\mathcal{L} = -\sum_{t=1}^{L} \log P(w_t|w_{t-1}, w_{t-2}, ..., w_1), \tag{1.2.3}$$

Neural Network LM Figure 1.1 gives the overview of the neural network language model [**7**]. Figure 1.1(a) is a simple neural language model architecture, which sum up the

embedding vectors [1] of input words. Then a non-linear layer is added on top of it to predict the output (the next word). In general, the neural network language model makes the n-th order Markov assumption:

$$P(w_t) = P(w_t|w_{t-1}, ..., w_{t-(n-1)}) \tag{1.2.4}$$

Otherwise, if using the full previous context, the sum of the embedding vectors will be too generic and meaningless.

Recurrent Neural Network RNN language model uses the full previous context as shown in Figure 1.1(b). It reads the input words one by one, and combine the corresponding word embedding with the previous representation to produce the next representation. At the end, once all the input words have been used, the final representation is used as the one that represents the whole sentence. As the input sequence is processed sequencially, RNN is time-consuming. Besides, previous work has shown that it (e.g. LSTM) can only consider up to ~ 100 words, otherwise, the model will fail to capture the meaning of the sentence.

Transformer Recent models in NLP usually consist of multiple Transformer layers. In Chapter 3, Figure 3.1 gives detailed architecture of a Transformer layer. Transformer is based on self-attention mechanism. Self-attention allows the tokens in a sequent to pay a certain amount of attention to any other tokens, depending on a weight determined by the similarity between the token and other tokens. According to the attention weights, the representation of the token at the next layer is produced by aggregating the other token's information. It reads words all together instead of one by one as shown in Figure 1.1(c). Therefore, unlike recurrent neural network, self-attention mechanism does not naturally incorporate position information. Thus, the input of Transformer is the sum of the word embeddings and the position embeddings [**91**][2]. In the current research, transformer is widely used to produce contexualized representations of tokens and sentences.

Given different self-attention masks that decides for each position which other positions can be attended to, Transformer can work as a bi-directional language model, e.g. BERT, or a left-to-right generative language model, e.g. GPT. Transformer has much larger capacity than RNN of the same number of parameters, and it is now state-of-the-art framework in NLP tasks.

[1] A word embedding is a real-valued vector that encodes the meaning of a word such that the words that are closer in the vector space are expected to be similar in meaning.

[2] Position embedding converts a relative position to a vector, aiming at distinguishing the word appearing at different positions. For example, the words at the first position all add the same position vector, and words at the n-th position add another vector. These vectors for different positions can be fixed or learned.

1.2.2. Pre-training Methods for LM

All the following language models are based on Transformer, and they are pre-trained [3] on large-scale text corpus. The largest difference among them is the self-attention mask and accordingly training objective.

GPT [**65, 66**] Generative Pre-trained Transformer is a generative language model that applies left-to-right self-attention, i.e. a word can only attend to previous words. GPT applies auto-regressive training objective. It is widely used when fine-tuning for a generation task, e.g. dialogue generation.

BERT [**15**] Bidirectional Encoder Representations from Transformers is a bi-directional language model, where a word can attend to every word in the input. Because of the bi-directional self-attention, BERT cannot apply auto-regressive training objective. Otherwise, there will be information leak. Thus, BERT applies Masked Language Modeling (MLM), an auto-encoding objective. A certain percentage of the words in the input are masked, i.e. replaced by a special token [MASK]. Then, the model is required to predict which word the masked token is given the bi-directional context. BERT is usually used as a pre-trained encoder, and it has been shown to have superior performance in many natural language understanding tasks. Nevertheless, we will show in Chapter 3 that fine-tuning BERT for generation task can also achieve state-of-the-art performance.

XLNet [**103**] is an extension of the Transformer-XL model pre-trained using an autoregressive method. It claims to incorporate all advantages of GPT and BERT. However, Roberta [**54**], a fully-trained BERT, has shown to outperform XLNet. Nevertheless, the motivation of XLNet is interesting, and thus we briefly introduce this approach here.

XLNet argues that BERT outperforms GPT since BERT considers bi-directional context. However, BERT using masked language modeling objective neglects dependency between the masked positions and suffers from a pretrain-finetune discrepancy [4] (because during fine-tuning, the input does not contain [MASK]). Therefore, the core idea of XLNet is not to use MLM objective but to use auto-regressive objective while considering bi-directional context. As mentioned above, MLM objective is necessary to avoid information leak when using bi-directional attention. Therefore, XLNet applies a trick: using auto-regressive objective and for each input text using all possible permutations of the factorization order. In expectation, each position learns to utilize contextual information from all positions, i.e. , capturing bidirectional context. To define a factorization order, they do not actually disorganize the input text. Instead, they apply a specific self-attention mask: by setting which tokens a given token can attend to, the self-attention mask then decides the position of the token.

[3]This process is named "pre-training" since people will further train, i.e. fine-tune, these language models on some other datasets.

[4]That is mismatch between the pre-training process and the fine-tuning process. We will discuss in detail pretrain-finetune and finetune-generation discrepancy in Chapter 3.

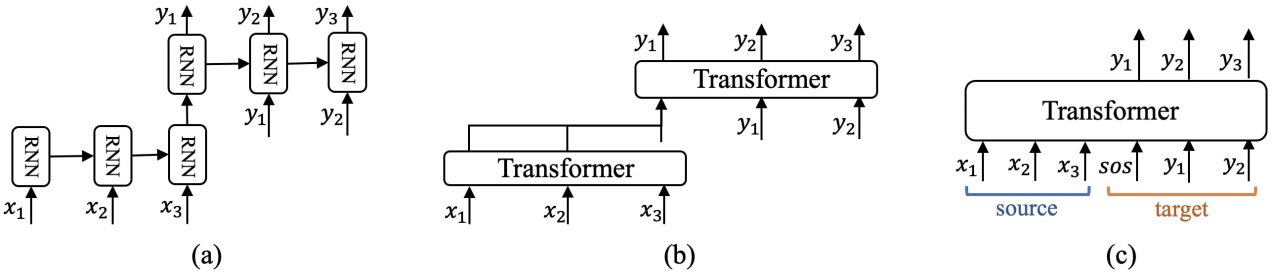

Fig. 1.2. (a) RNN based encoder-decoder framework; (b) Transformer encoder-decoder framework; (c) Transformer based decoder-only framework using different type embeddings (and self-attention masks) for the source side and the target side. For dialogue data, x_i is the i-th word of a dialogue history, and y_i is the i-th word of the ground-truth response. As mentioned before, "sos" represents "start of string", which is utilized to predict the first word in the response.

1.2.3. Sequence-to-Sequence Model

Encoder-Decoder Framework The encoder-decoder architecture encodes the input and then generates the output with decoder. There are two typical implementations: RNN-based and Transformer-based Seq2Seq models. For the former, the final hidden state of RNN encoder, a vector encoding the context of full input, is passed to the RNN decoder. For the latter, the output hidden states of Transformer encoder, $|L|$ vectors, are all passed into the Transformer decoder. Figure 1.2 (a) and (b) give an overview of these two frameworks. There are also frameworks consisting of Transformer encoder (BERT) and RNN decoder as we will introduced in Chapter 2.

Decoder-Only Framework This type of framework (see Figure 1.2 (c)) only uses a decoder to model a sequence-to-sequence task. We are not aware of RNN-based decoder-only framework probably because of the low capacity of RNN. In contrast, Transformer-based decoder-only framework has shown to have superior performance in many sequence-to-sequence tasks, including summarization [**53**] and dialogue generation (we will further introduce this framework in Chapter 3). For Transformer, an explicit encoder may be redundant, and we can concatenate the source and the target as the input of Transformer and apply different types of embedding to distinguish them. In Chapter 3, we will also show that when fine-tuning a pre-trained language model, a stacked encoder-decoder framework is less efficient, and performs worse than a decoder-only framework.

1.2.4. Transfer Learning Methods

Sequential Training (Pre-training and Fine-tuning) Pre-training and fine-tuning is of this type. The method starts from a pre-trained language model and sequentially, i.e. one by one, fine-tunes it on each of the tasks, i.e. datasets. However, when subsequently fine-tuning the model weights on new tasks, the problem of catastrophic forgetting [**58**] can

arise, which results in loss of knowledge already learned from all previous tasks. Besides, it is non-trivial to decide the order of tasks to be fine-tuned. In Chapter 4, we will explore sequential training in details.

Multi-task Learning In this case, the model is trained for several tasks simultaneously, i.e. combing all the datasets into one for training, with the aim of learning a shared representation that will enable the model to generalize well on each task.

However, multi-task learning requires access to all the tasks at the same time, making it difficult to add more tasks on the fly. Furthermore, it is difficult to balance multiple tasks and train a model that solves each task equally well. As has been shown in previous work [**41**], these models often overfit on low resource tasks and underfit on high resource tasks.

Adapter Recently, Adapter [**73, 33**] has emerged as a solution that overcomes the problems of catastrophic forgetting and imbalanced size of training sets. Adapter is a small set of task-specific parameters upon a fixed base model. This is different from the pre-training then fine-tuning paradigm, which modified the parameters of the whole model. Instead, an adapter will create a small adapter model on top of the base model. A typical adapter layer is a down projection (a linear layer) to a bottleneck dimension followed by an up projection (another linear layer) to the initial dimension. For multi-task learning, since Adapters for multiple tasks all share the same underlying base model, we can separately (then can be parallel) train them. Notice that we cannot train an adapter together with a pre-trained language model, but use the adapter upon another language model.

Many Adapter variants have been applied to diverse tasks including visual domain learning [**73**], language adaptation [**6, 63**], and knowledge infusion [**93**]. In Chapter 5, we will show that Adapter can be applied to build a multi-skill dialogue model.

In this chapter, we have described briefly the basic neural network architectures used in neural language modeling and dialogue systems. The description is intended to provide a background to understand more easily our studies on different aspects of dialogue described in the following chapters. We have not included all the technical details and refer the interested readers to the specific papers for them.

Chapter 2

Leveraging Pre-trained LM for Task-oriented Systems

This chapter presents methods to leverage pre-trained language model for Dialogue State Tracking (DST), a core component in task-oriented dialogue systems. DST aims at determining the current dialogue state once a user input an utterance. Figure 2.1 shows an example of dialogue state. The dialogue has some history (previous dialogue rounds), together with the previous dialogue state detected (S_1). Once the user inputs a new utterance (Usr in D_2), the goal of DST is to determine the dialogue state S_2. In this example, we can see that a dialogue state is composed of a set of triples (domain, slot, value). DST is critical to task-oriented dialogue which aims to help the user to accomplish a task such as booking a taxi or a restaurant. Only when the dialogue state is correctly determined can the task be accomplished correctly.

In previous works, BERT only worked as an encoder, and an RNN decoder is then stacked on it. Differently, we propose to use a single BERT model as both the encoder and the decoder [1]. It also has a flat encoder-decoder architecture allowing for more effective parameter updating. Experimental results show that our approach achieves state-of-the-art performance, and can converge to its best performance much faster and in a more stable manner than the existing framework. Furthermore, as discussed in Chapter 1, such a generative framework can be extended to build a task-oriented dialogue system if employing the idea of "text-to-text" to incorporate the other two tasks of task-oriented dialogue.

2.1. The Challenges of Dialogue State Tracking

DST is a core component in task-oriented dialogue systems. It aims at estimating the user's goal behind his utterance as the dialogue progresses. Accurate DST is crucial for appropriate dialogue management, where the user intention is an important factor that

[1]The code is available at `https://github.com/zengyan-97/Transformer-DST/`

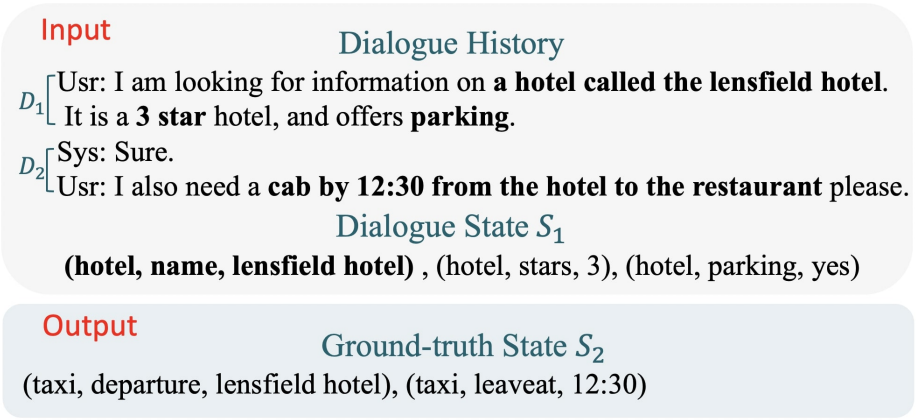

Fig. 2.1. An example of multi-domain DST.

determines the next system action. There are three challenges to build a DST model. First, the conversation can relate to multiple domains, e.g. restaurant booking and taxi. Second, we cannot pre-define slot values, e.g. the time when a taxi leaves a place, and thus an extractive or generative model is necessary to determine the slot value dynamically. Third, DST is not a genuine generative task, and we need to modify the generative framework to better adapt to this task.

Figure 2.1 shows an example of multi-domain DST [**8**], where the goal is to predict the *dialog state*, i.e. (*domain*, *slot*, *value*) tuples given the *dialogue history* and previous *dialog state*. The state S_1 corresponds to the utterance D_1. Given the new user's utterance in D_2, we aim at generating S_2. This example is from the MultiWoz 2.0 and 2.1 datasets, which will be used in our experiments.

Investigations on DST have started with ontology-based DST [**31, 59**], where all slots and possible values are predefined in the ontology. However, it is often difficult to obtain a large ontology in a real scenario [**101**]. Thus, recent studies focus on the open-vocabulary setting [**25, 107, 99**], where the possible *values* are not pre-defined and need to be directly extracted / generated from the input. The existing state-of-the-art generative framework [**37, 118, 105**] decomposes DST into two sub-tasks: State Operation Prediction (SOP) and Value Generation (VG). First, the encoder reads the dialogue history and previous dialogue state, and decides whether the value of a (domain, slot) pair needs to be updated. Then, the decoder generates a slot value for the (domain, slot) pair. This approach employs BERT [**15**] as the encoder and stacks an RNN-based decoder upon BERT outputs. By exploiting BERT as the encoder, this approach has substantially outperformed previous RNN-only framework (RNN encoder and decoder) [**99**], and achieved new state-of-the-art performance.

However, in this generative framework, the two sub-tasks of DST, i.e. state operation prediction and value generation, are not jointly optimized, which may lead to a sub-optimal model. Specifically, the SOP objective only affects the BERT encoder, while the VG objective

mainly affects the RNN decoder since the stacked encoder-decoder structure makes it less effective in updating the BERT encoder parameters [**53**]. Besides, in the framework, the encoder is pre-trained while the decoder is not. This second problem has been observed in previous work [**37**], and the proposed solution is to employ two different optimizers for the encoder and the decoder in the training process. The solution further separates the encoder and the decoder, or the SOP and VG process, making it even more difficult to make a global optimization. We will solve these problems in our work.

Third, when directly employing a powerful generative framework to DST, we observe, however, that the model performance drops sharply. The possible reason is that DST is not a genuine generation task that usually needs to cope with the entire input. For example, generating a dialogue response needs to be consistent with the dialogue history, or translating a sentence usually needs to translate each word on the encoder (input) side. In contrast, in DST, the value to be generated is only related to a very small fraction of the model input (within dialogue history and previous dialogue state) that usually consists of one or a few tokens. In this case, asking the decoder to take into account all the encoder outputs may blur the focus.

To solve this problem, we make the following adaptation by borrowing ideas from the existing state-of-the-art framework of DST: For a specific (domain, slot) pair, our decoder only re-uses the hidden states of the most relevant inputs. After an exhaustive search, our experiments show that re-using dialogue of the current turn and the slot state for the specific (domain, slot) pair yields the best performance, which substantially outperforms the previous framework and only needs a half of the training iterations.

2.2. Related Work

Traditional DST approaches rely on ontology. They assume that the possible values for each slot are pre-defined in an ontology and the problem of DST can be simplified into a value classification/ranking task for each slot [**31, 59, 112, 75, 68**]. These studies showed the great impact of ontology on DST. A recent work [**83**] combining ontology and contextual hierarchical attention has achieved high performance on MultiWOZ 2.1. In real application situations, however, one cannot always assume that an ontology is available [**101, 99**]. In many cases, slot values are discovered through the conversation rather than predefined (e.g. taxi departure time).

Open-vocabulary DST addresses this problem: it tries to extract or generate a slot value from the dialogue history [**42, 25, 74**]. In this work, we focus on open-vocabulary DST. However, many of the existing approaches did not efficiently perform DST since they generate a value for each slot at every dialogue turn [**99**]. In contrast, some works [**74, 37**] used a more efficient approach that decomposes DST into two successive sub-tasks: state operation

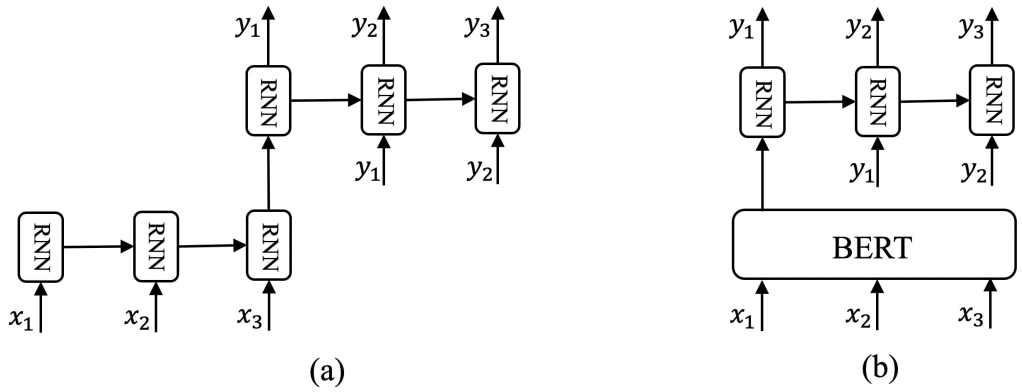

Fig. 2.2. (a) Overview of the RNN-based framework. (b) The SOTA generative framework that employs a BERT encoder and a RNN decoder. In both frameworks, the decoder is stacked upon the encoder outputs.

prediction and value generation. Many recent works [**118, 105**] are built upon this approach. However, these models do not jointly optimize the two sub-tasks, which may lead to a sub-optimal model since the performance of one process directly influences the performance of another process. For example, only when a slot indeed needs updating would generating a new value for it be meaningful. We will solve this problem in this work.

Studies on open-vocabulary DST have started with RNN-based encoder-decoder architecture as shown in Figure 2.2 (a). For example, previous work [**99**] encodes the dialogue history using a bi-directional GRU and decodes the value using a copy-based GRU decoder. Some recent studies have used pre-trained BERT as the encoder [**107, 74, 37**] to leverage the rich general linguistic features encoded in BERT as shown in Figure 2.2 (b). The existing state-of-art generative framework, SOM-DST[**37**], utilizes BERT as the encoder to predict state operations, and an RNN decoder stacked upon BERT to generate values. This gives rise to the separate training problems of the two processes we discussed earlier. Different from this approach, we use a single BERT as both the encoder and the decoder, and jointly optimize it with the SOP and VG objective. Furthermore, we also propose a flat encoder-decoder structure instead of stacked encoder-decoder, by re-using hidden states of the encoder in the self-attention mechanism of the corresponding decoder layers. This will lead to more effective updating of the encoder parameters [**53**].

Recently, Tripy [**30**], an extractive approach with 2 memory networks, and SimpleTOD [**32**], a language model, have also achieved high performance on MultiWoz 2.1. SimpleTOD adapts the idea of "text-to-text" [**66, 67, 36**] specifically to task-oriented dialogue, and applies multi-task learning on 3 subtasks of task-oriented dialogue including DST. DST can thus benefit from the two other sub-tasks. In addition, a much larger pre-trained language model (GPT-2) is used. In our work, we aim at developing a DST model that does not require a large amount of extra resource and can operate efficiently. This is because DST is

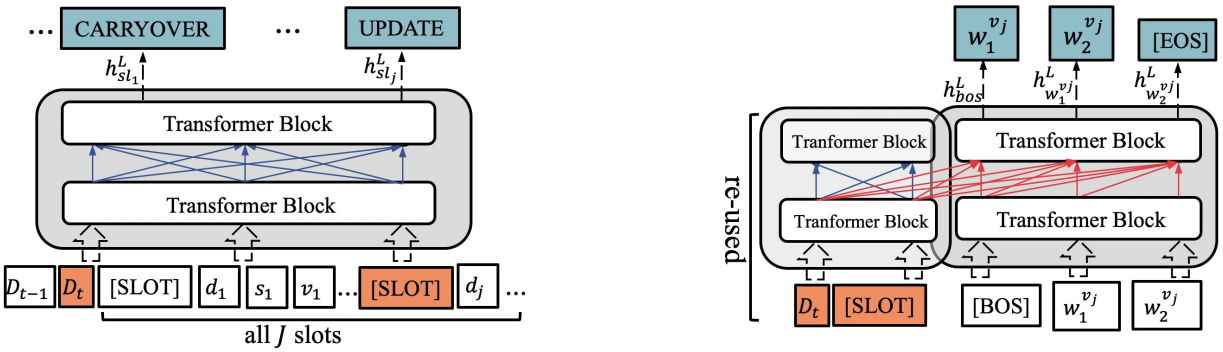

Fig. 2.3. (Left) The state operation prediction process, where the model (as the encoder) applies bi-directional self-attention mask. (Right) The value generation process for j-th (domain, slot) pair, where the model (as the decoder) applies left-to-right attention and re-uses the hidden states of the encoder in the corresponding decoder layers. The training objective is the sum of the state operation prediction loss and the value generation loss.

an intermediate task serving another end task (conversation). To be usable in real scenarios, the DST model should be both time- and memory-efficient.

2.3. Purely Transformer-based Framework

To solve the problems in previous works, we propose a purely Transformer-based framework for DST that exploits a single BERT as both the encoder and the decoder. When used as encoder, it processes state operation prediction as in previous works. When using it as decoder, we utilize different input representations to denote the target (decoding) side and left-to-right self-attention mask (i.e. attention is allowed only to previous positions) to avoid information leak. Therefore, the SOP objective and the VG objective affect both the encoder and the decoder, i.e. jointly fine-tuning this BERT for DST. Furthermore, instead of a stacked encoder-decoder structure as in previous studies, whose parameters cannot be effectively updated, we propose a flat structure by re-using the hidden states of the encoder in the self-attention mechanism of the corresponding decoder layers to make parameter updating in encoder more effective. Figure 2.3 gives an overview of our framework.

For multi-domain DST, a conversation with T turns can be represented as $(D_1, S_1), (D_2, S_2), ..., (D_T, S_T)$, where D_t is the t-th dialogue turn consisting of a system utterance and a user response, S_t is the corresponding dialogue state. We define S_t as a set of $(d_j, s_j, v_j)|1 \leq j \leq J$, where J is the total number of (domain, slot) pairs, i.e. S_t records slot values of all (domain, slot) pairs. If no information is given about (d_j, s_j), v_j is *NULL*. The goal of DST is to predict S_t given $\{(D_1, S_1), ..., (D_{t-1}, S_{t-1}), (D_t)\}$, i.e. we want to generate the state for the current turn t of dialogue, given the dialogue history and previous dialogue state. Following previous work [**37**], we only use D_{t-1}, D_t, and S_{t-1} to predict S_t.

2.3.1. State Operation Prediction

Encoder The input to the encoder is the concatenation of D_{t-1}, D_t, and S_{t-1}. Each (d_j, s_j, v_j) tuple in S_{t-1} is represented by [SLOT]$\oplus d_j \oplus - \oplus s_j \oplus - \oplus v_j$, where $\oplus$ denotes token concatenation, and [SLOT] and $-$ are separation symbols. Notice that s_j and v_j might consist of several tokens. As illustrated in Figure 2.3, the representations at [SLOT] position $\{\mathbf{x}^L_{sl_j} | 1 \leq j \leq J\}$ are used for state operation prediction. Then, we expect the hidden states at [SLOT] positions are able to aggregate the information from the corresponding (d, s, v) tuples. For example, each $\mathbf{x}^l_{sl_j}$ aggregates information of (d_j, s_j, v_j).

The input representation, i.e. $\mathbf{X}^0$, is the sum of token embedding, position embedding, and type embedding at each position. We apply type embeddings to introduce a separation between encoder side and decoder side. The multi-layer Transformer updates hidden states via: $\mathbf{X}^i = \text{Trans}^i(\mathbf{X}^{i-1}), \quad i \in [1,L]$. Specifically, within a Transformer Block, the multi-head self-attention mechanism is:

$$\mathbf{C}^l = \text{Concat}(\mathbf{head}_1, ..., \mathbf{head}_h) \tag{2.3.1}$$

$$\mathbf{head}_j = \text{softmax}(\frac{\mathbf{Q}_j \mathbf{K}_j^T}{\sqrt{d_k}} + \mathbf{M}^x)\mathbf{V}_j \tag{2.3.2}$$

where $\mathbf{Q}_j, \mathbf{K}_j, \mathbf{V}_j \in \mathbb{R}^{n \times d_k}$ are obtained by transforming $\mathbf{X}^{l-1} \in \mathbb{R}^{|x| \times d_h}$ using $\mathbf{W}^Q_j, \mathbf{W}^K_j, \mathbf{W}^V_j \in \mathbb{R}^{d_h \times d_k}$ respectively. The self-attention mask matrix $\mathbf{M}^x \in \mathbb{R}^{|x| \times |x|}$ (with $\mathbf{M}^x_{ij} \in \{0, -\infty\}$) determines whether a position can attend to other positions. Namely, $\mathbf{M}^x_{ij} = 0$ allows the i-th position to attend to j-th position and $\mathbf{M}^x_{ij} = -\infty$ prevents from it. In the state operation prediction process, $\mathbf{M}^x_{ij} = 0 \quad \forall i, j$.

Some hidden states of the encoder will be re-used in the decoder. The outputs of encoder are denoted as $\mathbf{X}^L = [\mathbf{x}^L_{cls}, \mathbf{x}^L_1, ..., \mathbf{x}^L_{sl_1}, ..., \mathbf{x}^L_{sl_J}, ...]$, which will be used for operation prediction.

Objective Following previous works [**25, 37**], we use four discrete state operations: CARRYOVER, DELETE, DONTCARE, and UPDATE. Based on the encoder outputs $\{\mathbf{x}^L_{sl_j} | 1 \leq j \leq J\}$, a MLP layer performs operation classification for each [SLOT]. Specifically, CARRYOVER means to keep the slot value unchanged; DELETE changes the value to NULL; and DONTCARE changes the value to DONTCARE, which means that the slot neither needs to be tracked nor considered important at this turn [**99**]. Only when UPDATE is predicted does the decoder generate a new slot value for the (domain, slot) pair.

2.3.2. Slot Value Generation

Decoder It applies different type embeddings to represent its input and left-to-right self-attention mask for generation to avoid information leak. The decoder re-uses [2] the hidden

[2]Re-using means the hidden states do not need to be calculated again in the decoder. In inference, the input of the decoder is only [BOS] to denote the beginning of the string.

states of encoder in the multi-head self-attention mechanism to construct a flat encoder-decoder structure making parameter updating in the encoder more effective:

$$\mathbf{Q}_j = \mathbf{Y}^{l-1}\mathbf{W}_j^Q \tag{2.3.3}$$

$$\hat{\mathbf{K}}_j = \text{concat}([\hat{\mathbf{X}}^{l-1}, \mathbf{Y}^{l-1}])\mathbf{W}_j^K \tag{2.3.4}$$

$$\hat{\mathbf{V}}_j = \text{concat}([\hat{\mathbf{X}}^{l-1}, \mathbf{Y}^{l-1}])\mathbf{W}_j^V \tag{2.3.5}$$

$$\mathbf{head}_j = \text{softmax}(\frac{\mathbf{Q}_j\hat{\mathbf{K}}_j^T}{\sqrt{d_k}} + \mathbf{M}^y)\hat{\mathbf{V}}_j \tag{2.3.6}$$

where $\mathbf{Q}_j \in \mathbb{R}^{|y|\times d_k}$, and $\hat{\mathbf{K}}_j, \hat{\mathbf{V}}_j \in \mathbb{R}^{(|\hat{x}|+|y|)\times d_k}$. $|\hat{x}|$ is the length of $\hat{\mathbf{X}}$ that is the re-used encoder hidden states . In the decoder, the self-attention mask matrix is $\mathbf{M}^y \in \mathbb{R}^{y\times(|\hat{x}|+|y|)}$ and we set $\mathbf{M}_{ij}^y = 0$ if $j \leq i$.

Note that the re-used hidden states of the encoder have already encoded the entire input to some extent because of the bi-directional attention applied. We will show in the experiments that re-using only the current turn of dialogue D_t and j-th [SLOT], i.e. $\mathbf{x}_{sl_j}^l, l \in \{1, L\}$, (if updating value for the j-th slot) achieves the best performance.

Objective The objective of the value generation process is the auto-regressive loss of generated slot values compared to the ground-truth slot values as in the previous works. We use teacher forcing all the time. The final training objective is the sum of the state operation prediction loss and the value generation loss.

2.4. Experiment Settings

2.4.1. Datasets

To evaluate the effectiveness of our approach, we use MultiWOZ 2.0 [**8**] and MultiWOZ 2.1 [**21**] in our experiments. These datasets introduce a new DST task – DST in mixed-domain conversations. For example, a user can start a conversation by asking to book a hotel, then book a taxi, and finally reserve a restaurant. MultiWOZ 2.1 is a corrected version of MultiWOZ 2.0. Following previous work [**37**], we use the script provided by previous work [**99**] to preprocess the datasets. The final test datasets contain 5 domains, 17 slots, 30 (domain, slot) pairs, and more than 4500 different values. Some statistics of MultiWOZ 2.1 are reported in Table 2.2. Table 2.1 shows the frequency of transitions from one domain to another in a dialogue (maximum 3 domains).

Domain Transition			
First	Second	Third	Count
restaurant	**train**	-	87
attraction	**train**	-	80
hotel	-	-	71
train	attraction	-	71
train	hotel	-	70
restaurant	-	-	64
train	restaurant	-	62
hotel	**train**	-	57
taxi	-	-	51
attraction	restaurant	-	38
restaurant	attraction	**taxi**	35
restaurant	attraction	-	31
train	-	-	31
hotel	attraction	-	27
restaurant	hotel	-	27
restaurant	hotel	**taxi**	26
attraction	hotel	**taxi**	24
attraction	restaurant	**taxi**	23
hotel	restaurant	-	22
attraction	hotel	-	20
hotel	attraction	**taxi**	16
hotel	restaurant	**taxi**	10

Table 2.1. Statistics of domain transitions that correspond to more than 10 dialogues in the test set of MultiWOZ 2.1. *Train* domain always co-occurs with another domain. *Taxi* always co-occurs with two other domains.

Domain	**Slots**	**Train**	**Valid**	**Test**
Attraction	area, name, type	8,073	1,220	1,256
Hotel	price range, type, parking, book stay, book day, book people, area, stars, internet, name	14,793	1,781	1,756
Restaurant	food, price range, area, name, book time, book day, book people	15,367	1,708	1,726
Taxi	leave at, destination, departure, arrive by	4,618	690	654
Train	destination, day, departure, arrive by, book people, leave at	12,133	1,972	1,976

Table 2.2. Data statistics of MultiWOZ 2.1 including domain and slot types and number of turns in train, valid, and test set.

2.4.2. Implementation Details

Our model is initialized with BERT (base, uncased), and it works as both the encoder and the decoder. We set the learning rate and warmup proportion to 3e-5 and 0.1. We use

a batch size of 16. The model is trained on a P100 GPU device for 15 epochs (a half of the iterations of SOM-DST). We use 42 as the random seed. With it, we can reproduce our experimental results. In the inference, we use the previously predicted dialogue state as input instead of the ground-truth, and we use greedy decoding to generate slot values.

2.4.3. Baselines

We compare the performance of our model, called Transformer-DST, with both ontology-based models and open vocabulary-based models.

FJST [**21**] uses a bi-directional LSTM to encode the dialogue history and a feed-forward network to choose the value of each slot.

HJST [**21**] encodes the dialogue history using an LSTM like FJST but utilizes a hierarchical network.

SUMBT [**40**] uses BERT to initialize the encoder. Then, it scores each candidate slot-value pair using a non-parametric distance measure.

HyST [**27**] utilizes a hierarchical RNN encoder and a hybrid approach to incorporate both ontology-based and open vocabulary-based settings.

DS-DST [**107**] uses two BERT-based encoders and designs a hybrid approach for ontology-based DST and open vocabulary DST. It defines picklist-based slots for classification similarly to SUMBT and span-based slots for span extraction as DST Reader.

DST-Picklist [**107**] uses a similar architecture to DS-DST, but it performs only predefined ontology-based DST by considering all slots as picklist-based slots.

DSTQA [**116**] formulates DST as a question answering problem – it generates a question asking for the value of each (domain, slot) pair. It heavily relies on a predefined ontology.

SST [**13**] utilizes a graph attention matching network to fuse utterances and schema graphs, and a recurrent graph attention network to control state updating.

CHAN-DST [**83**] employs a contextual hierarchical attention network based on BERT and uses an adaptive objective to alleviate the slot imbalance problem by dynamically adjust the weights of slots during training.

DST-Reader [**25**] formulates the problem of DST as an extractive question answering task – it uses BERT contextualized word embeddings and extracts slot values from the input by predicting spans.

DST-Span [**107**] applies BERT as the encoder and then uses a question-answering method similar to DST-Reader.

TRADE [**99**] encodes the dialogue history using a bi-directional GRU and decodes the value for each state using a copy-based GRU decoder.

NADST [**39**] uses a transformer-based non-autoregressive decoder to generate the current state.

	Model	BERT used	MultiWOZ 2.0	MultiWOZ 2.1
predefined ontology	HJST [**21**]		38.40	35.55
	FJST [**21**]		40.20	38.00
	SUMBT [**40**]	√	42.40	-
	HyST [**27**]		42.33	38.10
	DS-DST [**107**]	√	-	51.21
	DST-Picklist [**107**]	√	-	53.30
	DSTQA [**116**]		51.44	51.17
	SST [**13**]		51.17	55.23
	CHAN-DST [**83**]	√	**52.68**	**58.55**
open-vocabulary	DST-Reader [**26**]		39.41	36.40
	DST-Span [**107**]	√	-	40.39
	TRADE [**99**]		48.60	45.60
	COMER [**74**]	√	48.79	-
	NADST [**39**]		50.52	49.04
	SAS [**34**]		51.03	-
	SOM-DST [**37**]	√	51.72	53.01
	CSFN-DST [**118**]	√	52.23	53.19
	Craph-DST [**105**]	√	52.78	53.85
	Transformer-DST (ours)	√	**54.64**	**55.35**

Table 2.3. Joint goal accuracy (%) on the test set of MultiWOZ. Results for the baselines are taken from their original papers.

Model	Attr.	Hotel	Rest.	Taxi	Train
SOM-DST	69.83	49.53	65.72	**59.96**	70.36
Graph-DST	68.06	51.16	64.43	57.32	**73.82**
Ours	**71.11**	**52.01**	**69.54**	55.92	72.40

Table 2.4. Domain-specific joint goal accuracy on MultiWOZ 2.1.

SAS [**34**] uses slot attention and slot information sharing to reduce redundant information's interference and improve long dialogue context tracking.

COMER [**74**] uses BERT-large as the encoder and a hierarchical LSTM decoder.

SOM-DST [**37**] employs BERT as the encoder and a copy-based RNN decoder upon BERT outputs.

CSFN-DST [**118**] introduces the Schema Graph considering relations among domains and slots. Their model is built upon SOM-DST.

Graph-DST [**105**] introduces the Dialogue State Graph in which domains, slots and values from the previous dialogue state are connected. They instantiate their approach upon SOM-DST for experiments.

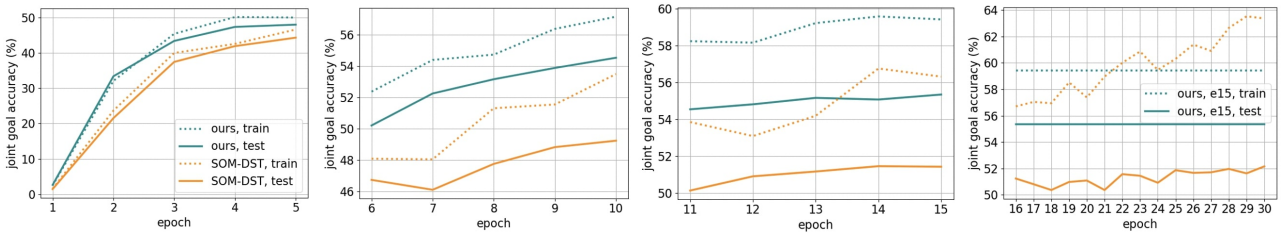

Fig. 2.4. The joint goal accuracy of Transformer-DST and SOM-DST on MultiWOZ 2.1.

2.5. Results

We report the joint goal accuracy of our model and the baselines on MultiWOZ 2.0 and MultiWOZ 2.1 in Table 2.3. Joint goal accuracy measures whether all slot values predicted at a turn exactly match the ground truth values. That is, a correct prediction should detect the right domain, slot and value. The joint goal accuracy is the rate of the correct detection over all the slots in the ground truth. The accuracy of baseline models is taken from their original papers.

As shown in the table, our Transformer-DST model achieves the highest joint goal accuracy among open-vocabulary DST: 54.64% on MultiWOZ 2.0 and 55.35% on MultiWOZ 2.1. Our model even outperforms all ontology-based methods on MultiWOZ 2.0. Recall that these latter benefit from the additional prior knowledge, which simplifies DST into a classification/ranking task.

In Table 2.4, we show the joint goal accuracy for each domain. The results show that Transformer-DST outperforms previous state-of-the-art framework (SOM-DST) in all domains except in *Taxi*. Graph-DST based on SOM-DST introduces the Dialogue State Graph to encode co-occurrence relations between domain-domain, slot-slot and domain-slot. This method outperforms our approach in *Taxi* and *Train* domains. According to the statistics about domains, we can see that *Taxi* and *Train* frequently co-occur with other domains. Therefore, leveraging extra knowledge about co-occurrence relations is particularly helpful for these domains. Our Transformer-DST does not exploit such knowledge that, however, could be added in the future.

In the following sub-sections, we will examine several questions: 1) how does joint optimization help the model to converge fast? 2) what is the model efficiency? 3) what is the impact of re-using different parts of the model input?

2.5.1. Joint Optimization Effectiveness

Figure 2.4 shows the joint goal accuracy on training set (5k samples) and test set of each epoch. We train Transformer-DST for 15 epochs and SOM-DST for 30 epochs as suggested in the original paper. We can observe that our model performance increases faster than SOM-DST at the beginning, and from 5th to 15th epoch the increase rate of the two frameworks

Model	Accuracy	Latency
TRADE	45.60	450ms
NADST	49.04	35ms
SOM-DST	53.01	50ms
Transformer-DST (Ours)	55.35	210ms

Table 2.5. Average inference time per dialogue turn on MultiWOZ 2.1 test set.

are close. At about 15th epoch, our performance generally stops increasing on both training set and test set.

Our model achieves 54% joint goal accuracy on the test set at 9th epoch, which already outperforms SOM-DST after 30 epochs. In contrast, SOM-DST does not outperform our model (15th epoch) on the training set until 22th epoch. On the test set, SOM-DST performance increases very slowly and is consistently worse than ours. These results suggest that SOM-DST at 30th epoch suffers more from the over-fitting problem than our model. We also observe that its performance fluctuates at the end.

We also observe that both the training and test curves of our framework are smoother than SOM-DST. The same observation is also made on MultiWOZ 2.0. This indicates that our training process is more stable and robust.

2.5.2. Inference Efficiency Analysis

As we have shown that our approach needs much fewer training iterations to achieve state-of-the-art performance, we further analyze its time efficiency at inference/test time. We show in Table 2.5 the latency of our method and some typical models measured on P100 GPU with a batch size of 1. Since our approach first predicts state operation, it is about 2 times faster than TRADE that generates the values of all the (domain, slot) pairs at every turn of dialogue. However, Transformer-DST utilizes a multi-layer Transformer (12 layers) for decoding, which makes it 4 times slower than SOM-DST. Overall, when latency is a critical factor in an application, it may be better to use SOM-DST or even NADST (using non-autoregressive decoder). In other cases or if we have a fast GPU device, the gain in accuracy of Transformer-DST is worth the higher cost in time.

2.5.3. Required Resource Analysis

The goal of our study is to improve DST without incurring much increase in resources. In parallel, several recent studies have explored using much larger resources for DST. Namely, Tripy [**30**] and SimpleTOD [**32**] also achieve high performance on MultiWoz 2.1 with joint goal accuracy of 55.30% and 55.76% respectively. However, these high performances are obtained at the cost of much higher resource requirements. This is why we have not discussed

Model	Input Len	Params	Resource
Tripy	512	BERT+2xMem	>2
SimpleTOD	1024	GPT-2	∼50
Ours	256	BERT	1

Table 2.6. Efficiency analysis of state-of-the-art approaches via comparing resource usage. BERT (base, uncased) we used has 110M parameters, while GPT-2 has 1.5B parameters.

them in previous subsections. Here, we compare our model with these two models in terms of cost-effectiveness.

Extra Knowledge Tripy uses auxiliary features. Without these extra features, Tripy only obtains 52.58% joint goal accuracy. SimpleTOD, a multi-task learning approach, also uses extra knowledge, i.e. supervision information from two other tasks closely related to DST. While our approach do not utilize any extra knowledge, the same extra knowledge could also be incorporated and would further boost the performance on DST.

Dialogue History Both Tripy and SimpleTOD utilize much longer dialogue history as model input, which is important for their models to achieve the state-of-the-art performance as reported. Encoding longer dialogue history is time-consuming and takes several times more GPU memories. To avoid this and to keep the whole process efficient, we utilize the predicted previous dialogue state as a compact representation of the dialogue history. Even with such a noisy input, our model can still achieve the state-of-the-art performance.

GPU Usage According to the input length and the amount of model parameters, we estimated the GPU resources (number of GPUs) needed by each approach as listed in Table 2.6. As mentioned, our approach only needs one P100 GPU for training. In contrast, although SimpleTOD is indeed simple, the GPT-2 it used is 13.6 times larger than the BERT we used. The large model is critical to this approach, as the authors also reported that the performance drops if a smaller pre-training model is used instead - with DistilGPT-2, still 8.6 times larger than our BERT, they only obtained 54.54% in joint goal accuracy. Therefore, our approach requires much less computation resources.

2.5.4. Re-Use Hidden States of Encoder

In our preliminary experiments, we re-use all hidden states of the encoder in the decoder, and the model performance drops sharply comparing to SOM-DST. Since DST is not a genuine generation task such as dialogue response generation that requires to be consistent with the entire dialogue history or machine translation in which every word on the source side needs to be translated, we consider re-using only a fraction of the hidden states. In SOM-DST, the RNN decoder only uses $\mathbf{x}_{cls}^{L}$ (the final hidden state at the first position) as the summary of the entire model input and $\mathbf{x}_{sl_j}^{L}$ (the final hidden state at the j-th [SLOT] position) as the summary of the j-th (domain, slot, value). Inspired by this, we conduct

Transformer-DST	Joint Accuracy
Full re-use	27.83
D_{t-1}+D_t+[SLOT]	53.08
D_t+[SLOT]	**55.35**
[CLS]+[SLOT]	53.95
[SLOT]+(d,s)	53.83
[SLOT]	53.03
D_t+[SLOT]+(d,s,v)	52.67
D_t+[SLOT]+(d,s)	52.40

Table 2.7. Joint goal accuracy on MultiWOZ 2.1 by re-using different encoder's states in the decoder.

an exhaustive search on which hidden states should be re-used. The results are listed in Table 2.7. We can see that re-using encoding hidden states of the current dialogue turn D_t and j-th [SLOT] achieves the best performance. Re-using more encoding hidden states may introduce additional noises.

2.6. Conclusion and Future Work

The existing state-of-the-art generative framework to DST in open-vocabulary setting exploited BERT encoder and copy-based RNN decoder. The encoder predicts state operation, and then the decoder generates new slot values. However, the operation prediction objective affects only the BERT encoder and the value generation objective mainly influences the RNN decoder because of the stacked model structure.

In this chapter, we proposed a purely Transformer-based framework that uses BERT for both the encoder and the decoder. The operation prediction process and the value generation process are jointly optimized. In decoding, we re-use the hidden states of the encoder in the self-attention mechanism of the corresponding decoder layers to construct a flat encoder-decoder structure for effective parameter updating. Our experiments on MultiWOZ datasets show that our model substantially outperforms the existing framework, and it also achieves very competitive performance to the best ontology-based approaches.

Some previous works in DST has successfully exploited extra knowledge, e.g. Schema Graph or Dialogue State Graph. Such a graph could also be incorporated into our framework to further enhance its performance. We leave it to future work.

Despite the fact that our approach can achieve state-of-the-art performance, we can notice that the joint accuracy rates are around 55% on the two collections. This shows that DST is a difficult task. In particular, in the case of open-domain DST, the detection of slot values is very difficult. In the ideal case, the dialogue system should capture exactly what the user says in order to place an order. This is still far from the case with the current

approaches. Therefore, more investigations are required before the dialogue systems can be put into practical utilization.

Chapter 3

Leveraging Pre-trained LM for Generative Chatbots

RNN-based Seq2Seq model is known to generate bland responses, e.g. "I don't know what you mean". Recently, models based on Generative Pre-training (GPT) have shown to generate fluent and diverse responses. Further, some works have even fine-tuned BERT, a well-known bi-directional encoder, for dialogue generation tasks. In this chapter, we compare 4 frameworks proposed in the literature that utilize pre-trained language models for open-domain dialogue generation on 3 public datasets, each in large and small scale, and we analyze each framework based on the experimental results. Through extensive experiments, we observe pretrain-finetune discrepancy and finetune-generation discrepancy of each framework. Then, we propose two methods to reduce discrepancies, yielding improved performance. It is the first investigation that shows explicitly the phenomenon of model discrepancy and its impact on performance.

Notice that even though recent studies have explored pre-training dialogue models using large-scale Reddit/Twitter data [**1, 77**] (it is then straightforward to fine-tune the models for a specific dialogue task), in practice, there may not always be enough data for pre-training. In some cases, we still need to exploit a pre-trained LM. For example, some studies do further pre-training for dialogue based on a pre-trained LM [**109, 18, 4, 86**], and some studies that do multi-task learning (e.g. on dialogue and question answering) can only fine-tune based on a pre-trained LM [**50, 104**]. Therefore, understanding how a pre-trained model can be best used in a dialogue task is crucial.

To give a better idea of what a chatbot system is required to generate, we provide a simple example in Table 3.1. The chatbot receives a user's utterance denoted as **Dialogue History**, and is asked to generate a response. The gold response in the dataset (the one given by another human being) is **Gold Response**. Notice that some of the tokens in the input and gold response have been replaced by their types (e.g. <num> and <person>) because it is difficult for a model to generate automatically the specific value for them. We

Dialogue History	one week before election day , early voting is nearly twice as high as <num>
Gold Response	i hope it 's <person> out in full force .
Trans-ED	i am not sure what you are talking about , but it 's a good thing that <person> is going to win .
Trans-Dec	that 's not true . early voting in nyc is times higher than the national average
Trans-MLM	it 's not even close to <num> % of the vote . i am sure you are right , but there is a huge difference between early voting and <person> voter suppression in ca
Trans-AR	it 's not that high , but i am sure there will be a lot of voter fraud .

Table 3.1. Sample responses generated by a chatbot.

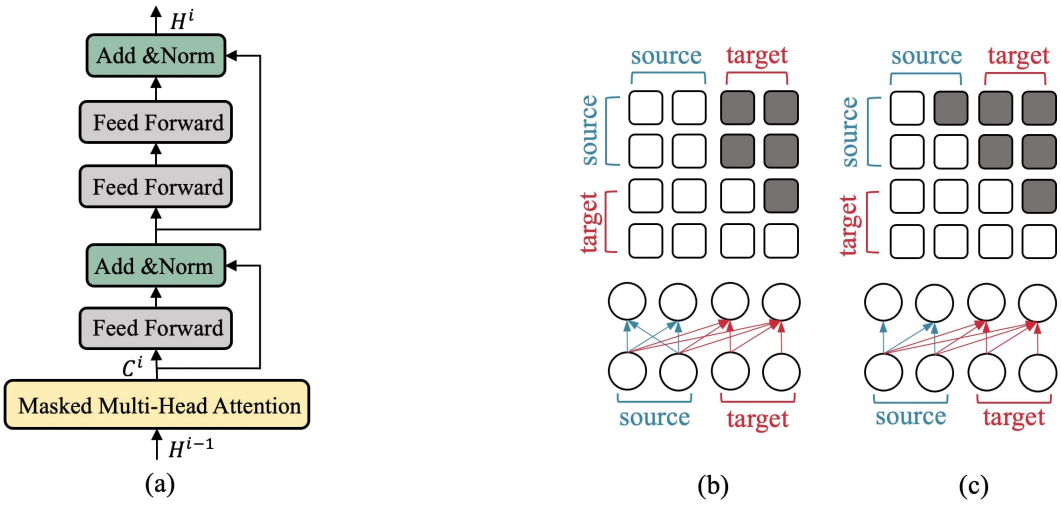

Fig. 3.1. i-th Transformer Block and two **M** settings represented in two ways. Shaded areas are blocked.

show the responses generated by four common models: Trans-ED, Trans-Dec, Trans-MLM and Trans-AR (which will be described in more details later). The quality of the response generated by a model is measured according to how it corresponds to the gold response (using automatic metrics such as BLEU), or using a manual evaluation to rate how reasonable the response is.

3.1. Multi-Layer Transformer

In this section, we recall some background knowledge on Transformer. We will introduce four frameworks that all consist of 12 Transformer blocks: Transformer-ED, Transformer-Dec, Transformer-MLM and Transformer-AR. More details about them will be introduced later. Figure 3.1 (a) shows a general architecture of a Transformer layer, where the most important component is the masked multi-head self-attention. The setting of attention masks is the largest difference between Transformer-Dec and Transformer-AR (which will

be detailed later) and it is also the most critical part to implement our discrepancy-free methods.

A dialogue history is denoted by x, and a corresponding response is denoted by y. The input to the multi-layer transformer is the concatenation of dialogue history and the response. When using MLM objective, the response is randomly masked. The input representation $\mathbf{H}^0 \in \mathbb{R}^{n \times d_h}$, where n is the input length and d_h is the hidden dimension (which is set at 768 in our experiments), is the sum of token embedding, position embedding, and type embedding at each position. The type embeddings introduce a separation between encoder/source side and decoder/target side in order to warrant different treatments in the model. Then, $\mathbf{H}^0$ is encoded into hidden representations of i-th layer $\mathbf{H}^i = [\mathbf{h}_1^i, ..., \mathbf{h}_n^i]$ by: $\mathbf{H}^i = \text{Trans}^i(\mathbf{H}^{i-1}), \quad i \in [1, L]$, where Trans^i denotes the i-th Transformer Block as shown in Figure 3.1(Left). The core component of a transformer block is the masked multi-head attention, whose outputs are $\mathbf{C}^i = [\mathbf{c}_1^i, ..., \mathbf{c}_n^i]$ that are computed via $\mathbf{C}^i = \text{Concat}(\mathbf{head}_1, ..., \mathbf{head}_h)$, with

$$\mathbf{head}_j = \text{softmax}(\frac{\mathbf{Q}_j \mathbf{K}_j^T}{\sqrt{d_k}} + \mathbf{M})\mathbf{V}_j \tag{3.1.1}$$

where $\mathbf{Q}_j, \mathbf{K}_j, \mathbf{V}_j \in \mathbb{R}^{n \times d_k}$ are obtained by transforming $\mathbf{H}^{i-1} \in \mathbb{R}^{n \times d_h}$ using $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V \in \mathbb{R}^{d_h \times d_k}$ respectively. $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the **self-attention mask matrix** that determines whether a position can attend to other positions. $\mathbf{M}_{ij} \in \{0, -\infty\}$. In particular, $\mathbf{M}_{ij} = 0$ allows the i-th position to attend to j-th position and $\mathbf{M}_{ij} = -\infty$ prevents from it. Figure 3.1 (b&c) shows two $\mathbf{M}$ settings that are applied by Trans-MLM/AR and Trans-Dec respectively.

3.2. Pre-training Based Frameworks

It has been shown that leveraging a pre-trained Language Model (LM) based on transformer can achieve excellent performance for dialogue generation [**97**]. Different approaches have been proposed recently, which can be categorized into 4 frameworks (see Fig. 3.2):

- Transformer-ED [**111**], an encoder-decoder Transformer;
- Transformer-Dec [**97, 50**], which uses Transformer only for decoder;
- Transformer-MLM [**18**], which uses Transformer with masked language model objective;
- and Transformer-AR [**4, 86**], which uses Transformer with autoregressive objective.

The latter three all utilize a decoder-only architecture. Besides, Trans-Dec uses left-to-right attention for both source and target side, while Trans-MLM and Trans-AR employ bi-directional attention on the source side to encode dialogue history. Due to this difference, Trans-Dec only utilizes left-to-right pre-trained models, e.g. GPT-2 [**66**], while Trans-MLM/AR are based on the pre-trained models applying bi-directional attention (on the

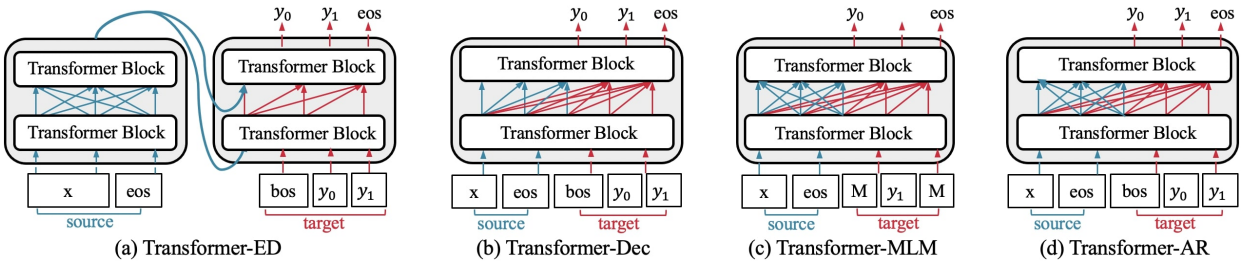

Fig. 3.2. Architectures of 4 pre-training based Transformers for dialogue generation.

	Trans-ED	**Trans-Dec**	**Trans-MLM**	**Trans-AR**
Pre-trained LM	GPT	GPT-2	BERT	BERT
Architecture	encoder-decoder	decoder-only	decoder-only	decoder-only
Source Side Attn.	bi-directional	left-to-right	bi-directional	bi-directional
Target Side Attn.	left-to-right	left-to-right	left-to-right	left-to-right
Objective	auto-regressive	auto-regressive	Masked-LM	auto-regressive

Table 3.2. Key characteristics of the 4 pre-training based Transformers. Characteristics in red are inconsistent between pre-training and fine-tuning.

source side), e.g. BERT [**15**]. The difference between Trans-MLM and Trans-AR is that Trans-MLM uses masked language modeling while Trans-AR uses auto-regressive objective.

Then, a critical question is how to best exploit a pre-trained LM for dialogue generation. On this question, we have contradictory beliefs in the literature: some researchers believe that Trans-Dec is appropriate because it uses a left-to-right language model that corresponds well to the dialogue generation task [**109, 50**], while some others [**18, 4**] show that Trans-MLM/AR fine-tuning BERT can also achieve state-of-the-art performance. We will explore this question through experiments in this work.

We start with a brief description of the 4 frameworks for dialogue generation based on pre-trained language models. We examine the pretrain-finetune discrepancy of each framework. Figure 3.2 and Table 3.2 provide an overview.

3.2.1. Model Discrepancy

The concept of model discrepancy has been briefly mentioned in XLNet [**103**] to mean that the model has been trained in a way, but used in a different way. However, the problem has not been investigated in depth. In this work, we go further in this direction and define two discrepancies: **pretrain-finetune discrepancy** which means the differences in architecture and loss function between pre-training and fine-tuning, and **finetune-generation discrepancy** which means that the way the model is used in generation (inference/test) is different from the way it has been trained. Discrepancies might affect the model performance since models with such discrepancies cannot best exploit the pre-trained model or employ

the fine-tuned model. For the 4 frameworks, except Trans-Dec, they all have some pretrain-finetune discrepancies. For example, Trans-AR relies on a BERT model pre-trained using bidirectional attention, but has to limit it to left-to-right attention on the target side during fine-tuning. Only Trans-MLM has finetune-generation discrepancy because of MLM objective: during training, the model input has random masks, while in the generation process, the input does not contain masks.

3.2.2. Transformer-ED

Trans-ED discussed in this paper is an encoder-decoder architecture used by ConvAI2 [**16**] champion [1]. The decoder of Trans-ED is stacked upon the encoder outputs, while in other decoder-only frameworks, the hidden states of each encoder layer are all utilized in the decoding part. The framework shares the encoder and the decoder and initializes the parameters with GPT [**65**]. In this case, the pretrain-finetune discrepancy comes from the bi-directional attention in encoder since GPT is a left-to-right language model. This framework is not commonly used for fine-tuning on a dialogue task. In practice, more efficient variants of Trans-ED are recently used for extremely large-scale dialogue pre-training from scratch. For example, some work [**1**] utilizes Evolved Transformer to prune redundant connections, and some [**77**] employs only 2 encoder layers and 24 decoder layers of standard Transformer [**91**].

3.2.3. Transformer-Dec

Trans-Dec is a left-to-right decoder-only architecture, and it utilizes GPT-2 [**66**]. Thus, there is no pretrain-finetune discrepancy in terms of architecture and loss function. This framework is widely applied for fine-tuning on a dialogue task. However, it encodes dialogue history using only left-to-right attention, which limits the scope of context, resulting in a partial context modeling.

3.2.4. Transformer-MLM and AR

These two frameworks have an identical decoder-only architecture [2] that employs different self-attention masks for the source and target side: they use bi-directional attention on the source side to encode dialogue history and left-to-right attention on the target side. The only difference between them is the objective function: Trans-MLM masks some tokens at the target side and tries to predict them, while Trans-AR uses auto-regressive objective that tries to predict the next tokens successively. BERT is often exploited by the two frameworks, which is a bi-directional architecture using MLM as the pre-training objective. Thus,

[1] `https://github.com/atselousov/transformer_chatbot`
[2] Since the architecture is not stacked encoder-decoder, we categorize it into decoder-only.

	Twitter	Ubuntu	Reddit
Train Set	2M	1.5M	3M
Valid Set	60K	30K	80K
Test Set	20K	20K	20K

Table 3.3. Key characteristics of the three public datasets. For each dataset, we also evaluate model performance using 100K training data and the same test set.

the pretrain-finetune discrepancy of Trans-MLM/AR comes from the left-to-right attention on the target side. Additionally, Trans-AR applies the auto-regressive objective, which is different from the MLM used in the pre-training.

3.2.5. Applications of the Frameworks

The frameworks we described have been widely applied to dialogue generation. For personalized response generation, some [**97**] uses Trans-Dec and some [**111**] utilizes Trans-ED. Some [**51**] uses Trans-Dec for empathetic response generation. Some [**104**] proposes a multi-task learning approach based on Trans-MLM for conditioned dialogue generation. Meanwhile, some studies propose to further pre-train the model using large-scale dialogue data based on a pre-trained language model: Some [**109**] trains Trans-Dec on 147M Reddit data based on GPT-2, some [**18**] trains Trans-MLM on natural language understanding and generation datasets based on BERT, some [**86**] trains Trans-AR on large-scale Reddit data and then jointly trains on 12 dialogue sub-tasks based on BERT, and some [**4**] trains a variant of Trans-AR on large-scale Reddit and Twitter data based on BERT. Some recent studies have increased the model size to billions of parameters and utilize even more training data, e.g. Reddit, to train a conversational model from scratch. Some works [**1, 77**] use variants of Trans-ED and some [**5**] employs a variant of Trans-AR.

In general, these studies show that all the 4 frameworks can produce good results, and increasing the model size and training data is an effective method to further improve performance. However, behind the success story, the question of suitability of a framework is overlooked. To investigate this question, we do not follow the current trend to increase the model size and training data. Instead, we are interested in the behaviors of different frameworks on the same datasets and to understand the reasons.

3.3. Experiments and Results

3.3.1. Datasets

We use three large-scale unlabeled dialogue datasets. Some important characteristics of the datasets are summarized in Table 3.3. We are interested in the behaviors of the models in two cases: 1) further pre-training on large dialogue data based on a pre-trained LM;

Model	Pre-trained LM	Data
Trans-ED	GPT [**65**]	BooksCorpus
Trans-Dec	GPT-2 small [**66**]	WebText
Trans-MLM/AR	BERT base [**15**]	BooksCorpus, English Wikipedia

Table 3.4. The text data used for language model pre-training.

Model	Params	Runtime
SEQ2SEQ-MMI	66M	50
HRED-MMI	58M	25
Trans-ED	117M	180
Trans-Dec	117M	290
Trans-MLM	110M	140
Trans-AR	110M	140
PF&FG-free	110M	140

Table 3.5. The number of parameters of each tested approach and the average runtime (minutes) for every million training samples. The runtime is tested using a 1080Ti GPU device, and the batch size is set to take all of the GPU memories. Notice that the runtime will be influenced by code implementation in addition to model structure.

and 2) fine-tuning on a small dialogue corpus based on a pre-trained LM. Our large datasets contain a few million samples, and the small datasets consist of 100K samples[3]. Although the datasets are smaller than those used in several previous studies, we believe that a comparison of different models on the same data, and the contrast between large and small datasets, can reveal interesting trends, which we will explain with respect to discrepancies.

Specifically, we choose the following 3 datasets: **Twitter Dialogue Corpus** [4] is collected from Twitter consisting of 2.6M (message, response) pairs. We filtered out samples with dialogue history (/context) length longer than 72 words (i.e. the previous rounds of dialogue longer than 72 words are filtered out due to limit the computation) or shorter than 6 words (not enough information). Samples whose response is longer than 36 words or shorter than 6 words are also removed. As a result, 2M samples are kept. **Reddit Conversational Corpus** [5][**20**] is a 3-turn conversational dataset collected from 95 selected subreddits. **Ubuntu Dialogue Corpus V2.0** [6] [**55**] contains two-person conversations extracted from the Ubuntu chat logs of technical support for various Ubuntu-related problems.

3.3.2. Implementation Details

We use open-source implementations for all four frameworks. Only minor adaptations (e.g. for data loading) have been made. We used the default settings for hyper-parameters, e.g. optimizer and learning rate. Although some models (e.g. Trans-ED) produced poor performance on small datasets, all model can generate some coherent and fluent responses with large scale training data, which is consistent with the performances reported in previous papers. The pre-trained language models used by these frameworks in previous studies have comparable number of parameters ($\sim$ 110M), while the pre-training data are in different scales: Trans-ED $<$ Trans-MLM/AR $<$ Trans-Dec. We assume that the difference is trivial when there are millions of dialogue data. Details are listed as follows.

Language Models The pre-trained language models used by these frameworks have comparable number of parameters as listed in Table 3.5, while the pre-training data are in different scales as described in Table 3.4. BooksCorpus [**119**] (800M words) contains over 7,000 unique unpublished books from a variety of genres. English Wikipedia (2,500M words) consists of the text passages of Wikipedia extracted by previous work [**15**]. WebText crawled by previous work [**66**] contains 8M diverse documents for a total of 40 GB of text.

Trans-ED We use the implementation of ConvAI2 champion [7]. The model was for persona-conditioned dialogue generation. The framework is based on GPT architecture and uses GPT for parameter initialization. However they only provide a model checkpoint that has been fine-tuned on large-scale dialogue data including Reddit. To examine the ability of utilizing pre-trained LM, we did not use this checkpoint but initialize the model with GPT parameters [8]. We also did not apply post-processing to the generation results (to be consistent with other experiments).

Trans-Dec We use the released code [**97**][9] that uses GPT-2 small by default. The model was for persona-conditioned dialogue generation.

Trans-MLM/AR These two models are implemented based on previous work [**18**] [10] that applies multi-task learning on language understanding and generation tasks. We use BERT (base, uncased) for parameter initialization, and fine-tune it on dialogue datasets. PF-free and FG-free are also implemented based on the code. We set the bi-directional attention interval of PF-free to 5. Since the average length of ground-truth responses in the datasets is $\sim$ 15, This setting is generally appropriate.

[3]Labeled datasets such as persona [**108**] and emotion [**71**] are usually in similar scale.
[4]`https://github.com/Marsan-Ma-zz/chat_corpus`
[5]`https://github.com/nouhadziri/THRED`
[6]`https://github.com/rkadlec/ubuntu-ranking-dataset-creator`
[7]`https://github.com/atselousov/transformer_chatbot`
[8]`https://github.com/openai/finetune-transformer-lm/tree/master/model`
[9]`https://github.com/huggingface/pytorch-openai-transformer-lm`
[10]`https://github.com/microsoft/unilm/tree/master/unilm-v1`

We also include two general RNN-based frameworks in this comparison to show how pre-trained models perform against them – **SEQ2SEQ-MMI** [**43**], a Seq2Seq model using bi-directional GRU encoder and applying Maximum Mutual Information (MMI) as the objective function to generate more diverse responses, and **HRED-MMI** [11], a hierarchical recurrent encoder-decoder neural network [**81**] applying diverse decoding strategy based on MMI [**45**].

We also equip all frameworks with an identical decoding script[12] to avoid extra factor affecting the generation quality, which uses beam search with beam size of 4, prevents duplicated uni-grams, and sets minimum response length that encourages diverse generation as in previous work [**77**]. The minimum response length is set to make the average length of generated responses match the average target length of the dataset. Generation results are evaluated after applying an identical word tokenization method. With two P100 GPU devices, the maximum input length is set to 128, and we fine-tune all models for 6 epochs and apply early-stop based on the performance on validation set. Our methods (PF-free and FG-free, which will be described in Section 3.4.2) do not add parameters or increase runtime in comparison with Trans-MLM.

3.3.3. Evaluation

Automatic Metrics We compare the similarity between generated responses and ground-truth responses using[13]: **BLEU** [**61**] evaluating how many n-grams (n=1,2,3) overlapped; **CIDEr** [**92**] utilizing TF-IDF weighting for each n-gram. Besides, we evaluate response diversity using **Distinct** (denoted Dist) [**43**] that indicates the proportion of unique n-grams (n=1,2) in the entire set of generated responses.

Human Evaluation Some existing studies considered response fluency, coherence, and informativeness. We make the manual evaluation simpler and ask two human evaluators who are professionals in dialogue systems to rate a response in $\{0, 1, 2\}$. A score of 0 represents an unacceptable response, which might have flaw in fluency and logic or be incoherent. Special cases are for example completely coping from the dialogue history as the output, and a bland response such as "i do not know what you are talking about , but it 's a good point .". A score of 1 represents an acceptable response, but it is generic or not perfectly coherent to the dialogue history. 2 represents a coherent and informative response. Each generated response is rated by three annotators. Annotators are unaware of which model generates a response. We also do a pair-wise evaluation to compare two models and indicate which one is better. To reduce time cost, we only perform human evaluation on Twitter and Reddit datasets that are closer to daily dialogue. However, during evaluation, we observe that $\sim 65\%$ Reddit data are professional discussions that are difficult to understand. The percentage is $\sim 30\%$ for

[11] `https://github.com/hsgodhia/hred`
[12] `https://github.com/microsoft/unilm/`
[13] We use an open-source evaluation tool: `https://github.com/Maluuba/nlg-eval`

Model	BLEU-1	BLEU-2	BLEU-3	CIDEr	Dist-1	Dist-2	avgLen
SEQ2SEQ-MMI	10.872 (**)	4.555 (**)	2.259 (/)	**0.119** (/)	0.008 (**)	0.028 (**)	10.6
Trans-ED	15.319 (**)	4.877 (**)	2.037 (**)	0.097 (**)	0.014 (**)	0.063 (**)	19.0
Trans-Dec	14.363 (**)	4.861 (**)	2.120 (*)	0.101 (**)	**0.031** (**)	**0.178** (/)	19.9
Trans-MLM	13.749 (**)	4.253 (**)	1.715 (**)	0.061 (**)	0.018 (**)	0.106 (**)	29.3
Trans-AR	**15.694**	**5.221**	**2.272**	**0.119**	0.029	0.164	18.9
FG-free	15.659 (/)	5.176 (/)	2.200 (/)	0.112 (/)	0.027 (**)	0.147 (*)	18.7
Trans-ED	14.813 (**)	4.249 (**)	1.330 (**)	0.066(**)	0.001 (**)	0.004 (**)	18.4
Trans-Dec	13.805 (**)	4.407 (**)	1.787 (**)	0.092(*)	**0.033** (**)	**0.195** (**)	20.2
Trans-MLM	15.487(**)	4.766(**)	1.814(**)	0.092 (*)	0.016(**)	0.080(**)	19.7
Trans-AR	15.213 (**)	4.700 (**)	1.767 (**)	0.090(**)	0.019(**)	0.091(**)	18.8
PF-free	15.880 (*)	4.970 (*)	1.868 (*)	0.093 (*)	0.022 (**)	0.114 (*)	15.7
FG-free	**16.395**	**5.218**	**2.043**	**0.101**	0.026	0.129	16.2
PF&FG-free	15.714 (*)	4.916 (*)	1.780 (**)	0.093 (*)	0.020 (**)	0.111 (*)	18.4

Table 3.6. Evaluation results on large-scale (upper half) and small-scale (lower half) Twitter dataset. PF-free denotes the method with reduced pretrain-finetune discrepancy of Trans-MLM. FG-free denotes the method that eliminates finetune-generation discrepancy of Trans-MLM. Two-sided t-test compares each method with the one without () sign, which is usually the best performer. Scores are denoted with * ($p < 0.05$) or ** ($p < 0.01$) for statistically significant differences.

Model	BLEU-1	BLEU-2	BLEU-3	CIDEr	Dist-1	Dist-2	avgLen
SEQ2SEQ-MMI	12.056(**)	5.512(**)	2.841(**)	0.142(**)	0.005(**)	0.024(**)	9.8
HRED-MMI	13.518(**)	4.564(**)	1.947(**)	0.060(**)	0.001(**)	0.003(**)	13.6
Trans-ED	19.295(/)	6.712(**)	2.986(*)	0.125(**)	0.010(**)	0.069(**)	16.8
Trans-Dec	18.974(*)	6.911(/)	3.022(*)	0.130(*)	**0.018**(**)	**0.134**(**)	18.0
Trans-MLM	17.574(**)	5.884(**)	2.552(**)	0.096(**)	0.012(**)	0.097(**)	25.5
Trans-AR	**20.103**	**7.270**	**3.339**	**0.143**	0.017	0.127	16.8
FG-free	19.774 (/)	7.045 (/)	3.213 (/)	0.139 (/)	0.016 (*)	0.115 (/)	17.7
Trans-ED	14.195(**)	4.533(**)	1.756(**)	0.074(**)	0.003(**)	0.012(**)	16.3
Trans-Dec	17.944(**)	6.360(*)	2.727(*)	0.121(/)	**0.018**(**)	**0.143**(**)	18.3
Trans-MLM	18.338(*)	6.018(**)	2.480(**)	0.108(**)	0.011(**)	0.066(**)	17.0
Trans-AR	19.005 (*)	6.431 (/)	2.733 (*)	0.114(*)	0.012(**)	0.078(**)	17.4
PF-free	**19.116** (*)	6.356 (*)	2.684 (*)	0.118 (/)	0.012 (**)	0.086 (*)	16.7
FG-free	18.884	**6.530**	**2.869**	**0.125**	0.014	0.095	17.3
PF&FG-free	19.024 (*)	6.448 (/)	2.740 (*)	0.118 (/)	0.012 (**)	0.087 (*)	17.1

Table 3.7. Evaluation results on large-scale (upper half) and small-scale (lower half) Ubuntu dataset.

Twitter data. These test samples are discarded, and at the end the test set for each dataset consists of 200 random samples. The inter-rater annotation agreement in Cohen's kappa [**14**] is 0.44 and 0.42 on average for Twitter and Reddit, which indicates moderate agreement (on the low side).

Model	BLEU-1	BLEU-2	BLEU-3	CIDEr	Dist-1	Dist-2	avgLen
SEQ2SEQ-MMI	15.550(**)	6.814(**)	3.321(**)	0.168(**)	0.011(**)	0.036(**)	11.2
HRED-MMI	13.278(**)	3.845(**)	1.398(**)	0.047(**)	0.001(**)	0.003(**)	13.8
Trans-ED	17.946(/)	6.626(**)	3.213(**)	0.165(**)	0.039(**)	0.203(**)	18.8
Trans-Dec	17.581(**)	6.790(*)	3.372(*)	0.180(**)	0.043(/)	**0.248**(**)	18.2
Trans-MLM	18.672(**)	7.115(**)	3.484(/)	0.177(**)	0.041(**)	0.215(**)	16.8
Trans-AR	**18.849**	**7.245**	**3.662**	**0.192**	**0.044**	0.235	16.8
FG-free	18.741 (/)	7.134 (**)	3.504 (*)	0.184 (*)	0.042 (**)	0.225 (**)	17.0
Trans-ED	17.337(**)	5.366(**)	1.967(**)	0.073(**)	0.001(**)	0.003(**)	17.1
Trans-Dec	17.460(**)	6.586(**)	3.161(*)	0.172(/)	**0.045**(/)	**0.254**(**)	17.7
Trans-MLM	19.193 (/)	6.877 (/)	3.175(*)	0.152(**)	0.029(**)	0.128(**)	15.0
Trans-AR	18.749(/)	6.746(/)	3.119(*)	0.153(**)	0.031(**)	0.141(**)	16.2
PF-free	18.466 (/)	6.688 (*)	3.075 (*)	0.169 (*)	0.038 (/)	0.180 (*)	14.1
FG-free	18.610	**6.937**	**3.302**	**0.175**	0.040	0.191	14.1
PF&FG-free	**19.302** (*)	6.923 (/)	3.073 (*)	0.159 (**)	0.034 (*)	0.164 (**)	15.3

Table 3.8. Evaluation results on large-scale (upper half) and small-scale (lower half) Reddit dataset.

Model	**Score** (M)	**Score** (K)
SEQ2SEQ-MMI	0.39	-
Trans-ED	0.53	0.11
Trans-Dec	**1.02**	0.77
Trans-MLM	0.88	0.58
Trans-AR	0.99	0.47
PF-free	-	0.52
FG-free	0.91	**0.78**
PF&FG-free	-	0.72
	Trans-Dec (M)	FG-free (K)
SEQ2SEQ-MMI	(11%, 48%)	-
Trans-ED	(14%, 46%)	(4%, 47%)
Trans-Dec	/	(24%, 29%)
Trans-MLM	(24%, 34%)	(18%, 31%)
Trans-AR	(27%, 32%)	(17%, 34%)
PF-free	-	(18%, 38%)
FG-free	(28%, 32%)	/
PF&FG-free	-	(23%, 29%)

Table 3.9. Human evaluation including pair-wise evaluation (lower half) for generated response quality for million-scale (M) Twitter dataset and its 100K training subset (K). Pair-wise comparison between two frameworks show the winning percentages of the two parties. For example, the first numbers (11%, 48%) mean respectively the percentage of the cases where Trans-Dec loses or wins against SEQ2SEQ-MMI. A higher winning rate than losing rate means that the method is better.

Model	**Score** (M)	**Score** (K)
SEQ2SEQ-MMI	0.12	-
Trans-ED	0.33	0.10
Trans-Dec	0.58	**0.43**
Trans-MLM	0.48	0.38
Trans-AR	0.64	0.31
PF-free	-	0.28
FG-free	**0.68**	0.40
PF&FG-free	-	0.33
	FG-free (M)	Trans-Dec (K)
SEQ2SEQ-MMI	(5%, 40%)	-
Trans-ED	(11%, 33%)	(2%, 28%)
Trans-Dec	(25%, 32%)	/
Trans-MLM	(18%, 29%)	(15%, 19%)
Trans-AR	(18%, 23%)	(15%, 23%)
PF-free	-	(15%, 24%)
FG-free	/	(23%, 24%)
PF&FG-free	-	(16%, 24%)

Table 3.10. Human evaluation on Reddit dataset.

3.3.4. Architecture Analysis

We first examine architecture appropriateness on the large-scale data setting, since when data are limited pretrain-finetune discrepancy and the size of pre-training data may strongly influence the results. Our global observation is that Trans-Dec and Trans-AR are the best choice for large-scale data setting, e.g. further dialogue pre-training based on a pre-trained LM.

Left-to-Right Only vs. Bi-Direction on the Source According to automatic metrics (Tables 3.6, 3.7, 3.8), Trans-AR applying bi-directional attention on the source side obtains the highest BLEU and CIDEr scores on all three million-scale datasets. We believe that bi-directional attention helps the model to better encode the dialogue history. In contrast, Trans-Dec is able to generate the most diverse responses. We attribute it to the left-to-right attention that introduces less constraint than bidirectional attention, thus has a higher flexibility for generation.

Human evaluation results in response appropriateness (Tables 3.9 and 3.10) show that Trans-Dec and Trans-AR generate most appropriate responses.

Trans-MLM vs. AR With large data, Trans-AR substantially outperforms Trans-MLM in terms of both automatic and human evaluation. When eliminating the finetune-generation discrepancy of Trans-MLM, i.e. FG-free (we will introduce in Section 3.4.3), the performance is improved while still having a small gap especially in automatic metrics to Trans-AR. This may be because MLM objective only masks a certain percentage of tokens (40%) while AR objective predicts all tokens on the target side for training. Thus, the AR

Dialogue History	one week before election day , early voting is nearly twice as high as <num>
Gold Response	i hope it 's <person> out in full force .
SEQ2SEQ-MMI	i am not voting for a third party candidate . (2)
Trans-ED	i am not sure what you are talking about , but it 's a good thing that <person> is going to win . (2)
Trans-Dec	that 's not true . early voting in nyc is times higher than the national average (2)
Trans-MLM	it 's not even close to <num> % of the vote . i am sure you are right , but there is a huge difference between early voting and <person> voter suppression in ca (2)
Trans-AR	it 's not that high , but i am sure there will be a lot of voter fraud . (2)
FG-free	it 's not even close to <num> yet , but i am sure the polls will show up . (2)
Trans-ED (K)	i am not sure what you are talking about . it 's the only one who is going to do with this (0)
Trans-Dec (K)	it 's not that early voting is a bad thing , but it shouldnot be the case . (0)
Trans-MLM (K)	it 's a little too late . i am not sure if that is the case , but we will see what happens next (2)
Trans-AR (K)	it 's not even close to <num> % of voter turnout . i am sure you are right about that (0)
PF-free (K)	<num> % of people voted for <person> , so i am going to vote (2)
FG-free (K)	it 's not even close to <num> % of voter turnout , but i am sure there are more . (2)
PF&FG-free (K)	it 's not even <num> % of the vote . i am going to be a voter (2)

Table 3.11. Responses generated by baselines and our methods trained on the Twitter dataset(million-scale and 100K). Human evaluation scores are given at the end of each generated reply.

objective is more training-efficient. However, when training data are limited, we will show that it is better to use MLM objective which has smaller pretrain-finetune discrepancy.

Trans-ED vs. Decoder-Only With large dialogue data, we assume the size of pretraining data only has small influence on performance. However, even comparing with Trans-MLM(FG-free)/AR, Trans-ED generates much less diverse or appropriate responses. We also observe lower speed for convergence when training the model [14]. We believe that the result is more or less due to the main difference in architecture: the decoder of Trans-ED only utilizes the outputs of the encoder, while all hidden states of the encoding part are used in other decoder-only frameworks. The different performances suggest that the latter architecture is a better choice.

In Tables 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, we show some examples generated by different models on the three datasets.

[14]Similar observation has been reported in: `https://github.com/atselousov/transformer_chatbot/issues/15`

Dialogue History	i think about this man every day
Gold Response	it is so hypnotic . this man was found dead post-election
SEQ2SEQ-MMI	i do not know what you are talking about . (0)
Trans-ED	i think he 's the only one who has a lot of work to do with it . (1)
Trans-Dec	i think about this man every day too . he 's my favorite person in the world (2)
Trans-MLM	he 's such a good guy . i love him so much, but it doesnot make sense to me that this is the only way we can be together in our lives (2)
Trans-AR	i think about this man every day too . he 's so good at it ! (2)
FG-free	he 's such a good guy . i am so happy for him and his family (2)
Trans-ED (K)	i am not sure what you are talking about , but it 's a good idea . (0)
Trans-Dec (K)	i think about this man every day . he 's my hero and i love him (2)
Trans-MLM (K)	i think he 's a great actor , but no one cares about him . lol (2)
Trans-AR (K)	i think he 's a good guy , but not sure what you are talking about . (0)
PF-free (K)	i think he 'snitch . it makes me happy to watch him (0)
FG-free (K)	he 's a great guy . i am so happy to have him back (2)
PF&FG-free (K)	i think he is little silly , but you are a fool . (2)

Table 3.12. Responses generated by baselines and our methods trained on the Twitter dataset(million-scale and 100K). Human evaluation scores are given at the end of each generated reply.

Dialogue History	how old are you guys ? you did not have mechanical pencils ? [SEP] <url> young are you ? i was in school all the way up to a few years ago and most of us still used non - mechanical ones .
Gold Response	i started using mechanical pencil in elementary school about 15 years ago .
SEQ2SEQ-MMI	i do not know what you are talking about . (0)
Trans-ED	how old are you ? i was in school all the way up to a few years ago . (1)
Trans-Dec	i was in school all the way up to a few years ago and most of us still used non - mechanical ones . (0)
Trans-MLM	i was in school for a few years , and most of us still use mechanical pencils . it 's not the same age as you (2)
Trans-AR	i was in high school when we used to use mechanical pencils . (2)
FG-free	how old are you ? i was in high school and still use mechanical pencils . (2)
Trans-ED (K)	i do not know what you are talking about , but it 's just a joke . (0)
Trans-Dec (K)	i was in school all the way up to a few years ago and most of us still used non - mechanical ones . (0)
Trans-MLM (K)	i have been in school for a few years , but not really . (0)
Trans-AR (K)	how old are you ? i have never been in school for a few years . (0)
PF-free(K)	how old are you ? i am not really a mechanical engineer . ! (1)
FG-free (K)	how old are you ? i have never had a mechanical pencil . (2)
PF&FG-free (K)	how old are you ? i have never had a lot of time in school (2)

Table 3.13. Responses generated by baselines and our methods trained on the Reddit dataset(million-scale and 100K). Human evaluation scores are given at the end of each generated reply. We can observe that some generated responses (in red) are simply copy from dialogue history.

Dialogue History	i have read all of the books , which character do you not like ? [SEP] it 's gon na be the dornish and greyjoys . and probably brienne . you will see .
Gold Response	nah , they have only just started a game of thrones so it must be one of the starks , dan , or tyrion .
SEQ2SEQ-MMI	i do not know why you are being downvoted . (0)
Trans-ED	i have read all of the books , and it 's gon na be dornish . (0)
Trans-Dec	i have not read the books yet , but i will definitely check them out . (2)
Trans-MLM	i have not read the dornish and greyjoys , but you will see them all as well . (0)
Trans-AR	brienne is not a fictional character , she 's an actual person . (2)
FG-free	i have read all of the dornish and greyjoys . (1)
Trans-ED (K)	i am not sure what you are talking about , but it 's just a joke . (0)
Trans-Dec (K)	i do n't know , i have read all of the dornish books . (0)
Trans-MLM (K)	i do not think it 's a good idea to say that . (0)
Trans-AR (K)	i do not think it 's gon na be the best . (0)
PF-free (K)	i am not sure what you are talking about here 's a link .
FG-free (K)	i do not think it 's gon na be . (1)
PF&FG-free (K)	i do not know , but it 's a lot of people ? (1)

Table 3.14. Responses generated by baselines and our methods trained on the Reddit dataset(million-scale and 100K). Human evaluation scores are given at the end of each generated reply.

Dialogue History	is that easy to do ? [SEP] usually the ubuntu installer will set it up perfectly for you
Gold Response	and then i would select at start-up ?
SEQ2SEQ-MMI	how do i install it ?
HRED-MMI	i am not sure what you mean by that
Trans-ED	is there a way to do it without installing the ubuntu installer ? i am not sure .
Trans-Dec	ok , i will try that . thanks for the help ! :-) it 's working now
Trans-MLM	is there a way to do it with the live cd ? i am not sure if that 's what you are looking for , but how does one install ubuntu on
Trans-AR	ok , i will give it a try . thank you very much for your help !
FG-free	is there a way to do it from the command line ? i am not sure what you mean
Trans-ED (K)	i am not sure what you mean by that , but it 's just a little bit more specific
Trans-Dec (K)	ok , i will try that . thanks for your help ! :-) <url>
Trans-MLM (K)	i am not sure what you mean by that , but it 's a good idea .
Trans-AR (K)	i am not sure what you are talking about , but it 's possible that the installer will be able to boot from cd
PF-free (K)	i am not sure what to do ? it ' s just a file in the repos
FG-free (K)	i do not think it ' s a good idea to install ubuntu on my laptop
PF&FG-free (K)	i am not sure what you mean by setting it up perfectly . is there a way to do that ?

Table 3.15. Responses generated by baselines and our methods trained on the Ubuntu dataset(million-scale and 100K).

3.3.5. Discrepancy Impact

In section 3.2, we have discussed the pretrain-finetune discrepancy of each framework. When a large training dataset is available, the impact of pretrain-finetune discrepancy is less

Dialogue History	usually if on the same machine , <url> [SEP] : i am getting error . phpmyadmin not in this server
Gold Response	you installed it from the software center , or with aptitude or apt-get , or some other way ?
SEQ2SEQ-MMI	what is the error you are getting ?
HRED-MMI	i have no idea what you are talking about
Trans-ED	i am not sure what you are trying to do , but it 's a good idea .
Trans-Dec	did you install phpmyadmin from the repos ? if so , it should be in <path>
Trans-MLM	i am not sure what you are trying to do , but it should be in <path> . try sudo dpkg - reconfigure phpmyadmin
Trans-AR	i am not sure what you are trying to do , but it 's possible .
FG-free	i am not sure what you are trying to do , but it ' s usually in <path>
Trans-ED (K)	i am not sure what you mean by " phpmyadmin <path>
Trans-Dec (K)	i am not sure what phpmyadmin is , but it 's probably in <path>
Trans-MLM (K)	i am not sure what you mean by " phpmyadmin - > <path> . <url>
Trans-AR (K)	i do not know what phpmyadmin is , but it 's in the repos
PF-free (K)	it ' s not in this server . i am trying to get phpmyadmin working
FG-free (K)	phpmyadmin is not in the same machine , it ' s just a server .
PF&FG-free (K)	i am not sure if it 's on the same machine , you can use phpmyadmin

Table 3.16. Responses generated by baselines and our methods trained on the Ubuntu dataset(million-scale and 100K).

severe since the model can be gradually adapted to the given task. However, if the training data are limited, the discrepancy problems may surface. Evaluation results, especially in human evaluation, show that the performance is more reduced with small data if the framework has larger discrepancy. For example, by comparing Trans-MLM (FG-free) and Trans-AR, the latter having additional pretrain-finetune discrepancy due to its auto-regressive objective, we see that the performance of Trans-AR drops more when trained on a small dataset. Trans-MLM (FG-free) and Trans-Dec that have small pretrain-finetune discrepancy have clear advantage over other frameworks according to human evaluation.

These results suggest that with a small dataset one should reduce pretrain-finetune discrepancy to best exploit pre-trained LM. In the next section, we propose 2 methods to reduce pretrain-finetune discrepancy and finetune-generation discrepancy of Trans-MLM.

3.4. Discrepancy-Free Transformer-MLM

3.4.1. The Attention Conflict Problem

If applying bi-directional attention at each generation step, the corresponding training method has extremely low sample-efficiency – only one token at the target side could be masked for each training sample; otherwise there will be attention conflicts, i.e. different self-attention mask matrices are required for different masked tokens, while only one mask matrix can be provided per training sample. We experimentally tested this approach and

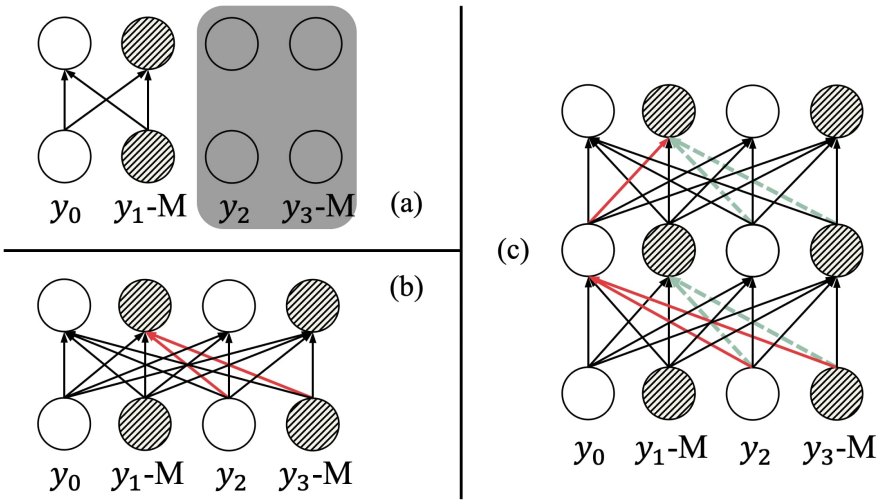

Fig. 3.3. Self-attention mask, $\mathbf{M}$, conflicts – (a) if predicting y_1, $\mathbf{M}^{(a)}$ is as the left figure, where y_2 and y_3-M are "future" and forbidden to be accessed by y_1-M; (b) if predicting y_3, $\mathbf{M}^{(b)}$ is as the right figure, in which case y_1-M accesses to y_2 and y_3-M. (c) if forbidding y_1-M to access to y_2 and y_3-M in $\mathbf{M}^{(b)}$, there will still be (indirect) information leak as indicated in red arrows. Masking two positions thus causes conflicts. Our PF-free method aims to overcome this problem.

found it less efficient and effective. In Figure 3.3, we provide an illustration of the mask conflict problem. We assume y_1 and y_3 are masked and need to be predicted at the same time. In the figure, we show two conflicting masks required for predicting y_1 and y_3. We see in the figure that two different masks are required for predicting y_1 and y_3, which cannot be done in a single training step, making it impossible to mask more than one token in each step.

3.4.2. Pretrain-Finetune Discrepancy

The discrepancy of Trans-MLM comes from the left-to-right attention on the target side that has not been pre-trained in BERT. Therefore, this discrepancy cannot be eliminated during fine-tuning for a generation task. However, we can alleviate the discrepancy by using bi-directional attention also on the target side. Specifically, at inference time, to generate a new token denoted as g_t, [MASK] is fed into t-th position, denoted as g_t-M. Previously generated tokens $g_{<t}$ could be viewed as a special type of dialogue history, and thus we can apply bi-directional attention on it.

However, in this case, the corresponding training process will have efficiency problems – only one token can be masked in each training sample; otherwise, there will be conflict for the self-attention mask. This would lead to much lower training efficiency: the loss on validation set only decreases slightly to 5.39 from 6.27 after four epochs, while Trans-MLM masking 40% of the target tokens can reduce it to 4.35. To avoid this situation, we cannot always

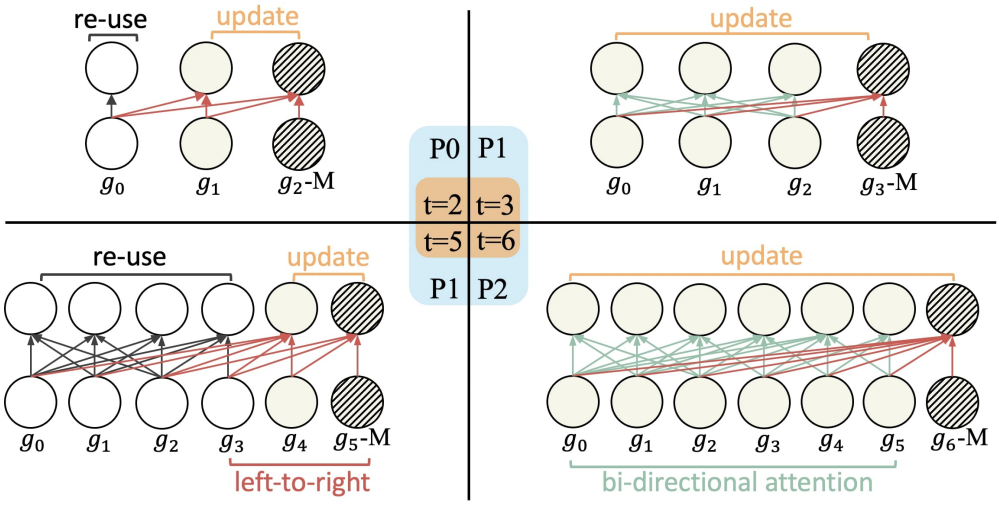

Fig. 3.4. The generation process of PF-free at 4 different time steps. Bi-attention interval is 3 in the graph.

update previous hidden states using bi-directional attention in generation. Therefore, we explore to set a time-step interval for bi-directional attention on the target side – within the interval we apply left-to-right attention and at the end of an interval we apply bi-directional attention. The corresponding training method allows us to mask multiple target tokens at the same time to guarantee training efficiency.

Figure 3.4 illustrates the generation process of our method with interval of 3. Before time step 3, left-to-right attention is used (e.g. t=2). At time step 3, bidirectional attention is allowed. Then left-to-right attention is used (e.g. t=5) before the end of next interval cycle (t=6). Accordingly, the training process is: given a target response, we first randomly select among all (3 in the figure because t=3 and t=5 are the same pattern) possible attention patterns (e.g. the case of t=3 or t=5 in Figure 3.4, where we apply bi-directional attention only on $y_{0,1,2}$); then in the part of left-to-right attention, we randomly mask several tokens. We can mask multiple tokens because this part applies left-to-right attention and the masks at other positions will not influence the prediction on a given mask. We call this method **PF-free**, which means that the pretrain-finetune discrepancy is reduced.

3.4.3. Finetune-Generation Discrepancy

A model having finetune-generation discrepancy means the way that it is used in generation (inference/test) is different from the way it has been trained. Only Trans-MLM has finetune-generation discrepancy because of its MLM objective as shown in Figure 3.5: during training, there is a masked token, y_1-M, before y_2-M, while in inference there is not a masked token before when generating the token for g_2-M.

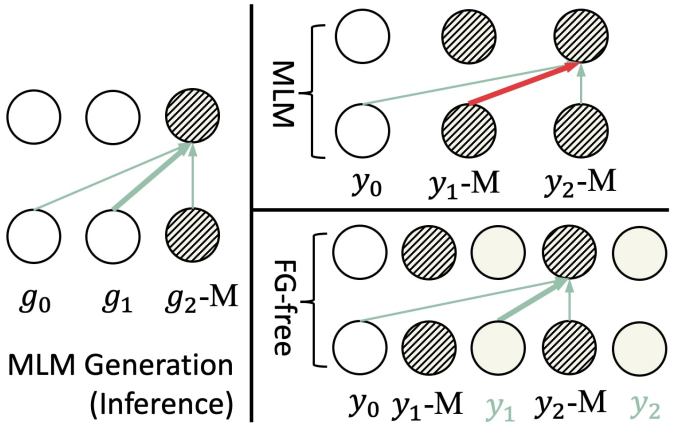

Fig. 3.5. The training process of vanilla Trans-MLM and FG-free. We only plot the attention connection at the second position.

To eliminate this mismatch, we propose that at training time, rather than replacing the tokens with [MASK] as in vanilla MLM, we keep all original input tokens unchanged and prepend [MASK] tokens in the input sequence as illustrated. In so doing, we can choose to use [MASK] or the original token according to the need. The prepended [MASK] token uses the same position embedding of the corresponding token. Then, every position after y_1-M attends to y_1 instead of the [MASK] token, and thus the finetune-generation discrepancy of MLM is eliminated as shown in Figure 3.5. We call the modified model **FG-free**. A similar method has also been explored in [**3**], where they introduced an extra pseudo mask in addition to [MASK] and prepend it before the original token in order to handle factorization steps of their partially auto-regressive language model.

3.4.4. Experimental Results

The results with PF-free, FG-free and PF&FG-free models on small-scale datasets are reported in Table 3.6, 3.7, 3.8, 3.9, and 3.10 together with other models. We can see that each of the proposed methods brings some improvement. PF-free improves most automatic metrics over Trans-MLM, but the response appropriateness in human evaluation is not improved. We observe that PF-free could generate some responses that lack fluency, which also influences PF&FG-free. In general, our exploration shows that the left-to-right attention on the target side is necessary for a generative task.

We examine our PG-free method on both large and small-scale data. It always brings statistically significant improvement over Trans-MLM in all automatic metrics, and generates more appropriate responses. On small-scale datasets, it outperforms all other frameworks in similarity metrics and achieve comparable performance in response appropriateness to Trans-Dec that has leverages much more pre-training data.

This set of experimental results confirm the usefulness of reducing discrepancies in the model. This demonstrates that model discrepancies are indeed important problems we need to address when a pre-trained LM is used for dialogue generation, and the problems have been under-explored.

3.5. Conclusion

In this chapter, we examined the 4 frameworks for open-domain dialogue based on pre-trained models. We compared their performances on several datasets with the same setting, each with large and small scale training data. Our results on large-scale datasets show that Transformer-ED that applies the stacked encoder-decoder architecture does not produce competitive results against the others that use a decoder-only architecture. Transformer-Dec/AR generate the most appropriate responses. However, according to automatic metrics, Transformer-Dec generates most diverse responses while Transformer-AR produce responses most similar to the ground-truth. This may be due to the fact that uni-directional attention does not have constraint from the right side context and thus is more flexible, while bi-directional attention on source side can better model dialogue context. In contrast, the results on small-scale datasets reveal an important aspect, namely, the discrepancies that may occur between pre-training and the fine-tuning processes. We then try to explain the performances of the 4 frameworks with respect to the discrepancies.

We defined the concept of pretrain-finetune and finetune-generation discrepancy, and examines the 4 frameworks with respect to these concepts. We have shown that the performances of the 4 frameworks can be largely explained by their respective discrepancies, which hinder their performances. This becomes more clear when the dataset is small.

To further show that reducing the discrepancies can improve the performance, we designed PF-free and FG-free correction methods to reduce the discrepancies on Transformer-MLM, and tested the corrected Transformer-MLM models on the datasets. Our results confirmed that once discrepancies are eliminated, Transformer-MLM can produce better results.

This study is the first investigation on the widely used 4 frameworks based on pre-trained LM in terms of architectural appropriateness and discrepancies. We believe that this question is important to understand how a pre-trained model can be used in dialogue generation. It deserves more investigations in the future.

Chapter 4

Multi-Task Learning based on Pre-trained LM for Conditioned Dialogue Generation

We have introduced the frameworks to leverage pre-trained language models for each type of dialogue systems. We will focus on conditioned dialogue for generative chatbot systems, i.e. dialogue that needs to fit a condition such as style, topic, etc. We take a very general view of conditioned dialogue to mean any dialogue that should fit some condition. The condition could denotes a specific person. In that case, we want the generated dialogue to mimic the style of that person. The condition can also be a specific topic domain (e.g. computer science). In that case, we aim to generate dialogues in that domain. Notice that this generalized view of conditioned dialogue is new. In all the existing work, one focuses on one single condition. We show in this chapter that a general approach can be developed to fit different types of condition.

In this chapter, we propose a simple and efficient multi-task learning approach based on pre-trained Transformer that leverages different condition-labeled data, i.e. dialogue and text, for conditioned response generation [1]. We assume that we have a set of conditioned dialogue data (i.e. dialogue relating to the condition) as in most of the existing studies. In addition, we also assume that we have a set of non-dialogue data corresponding to the condition (e.g. a set of texts written by a specific person), which can complement the former. The exploitation of the latter data is new in our work. The experiments under two different conditions – persona- and topic-based dialogue, show that our approach outperforms the state-of-the-art models by leveraging labeled texts even when the labels are predicted by a model.

[1]The code is available at `https://github.com/zengyan-97/MultiT-C-Dialog`.

4.1. The Challenge of Conditioned Generation

General conversational models pre-trained on large text data [**65, 15**] or human-to-human conversation data [**109, 4**] have shown excellent performance in generating fluent and diverse responses. In addition to general conversation, we are more and more faced with the problem of conditioned conversation that tunes the dialogue toward a specific style or domain. For example, we might specify a condition as the vocabulary frequently used by a person and ask the system to mimic the speaking style of the person, or a topic-related vocabulary and ask the chatbot to discuss the given topic. We put all these into the general category of conditioned dialogue and we aim at developing a general solution to such problems.

Conditioned response generation has been extensively explored using RNN-based sequence-to-sequence models, under different conditions, e.g. persona [**44**], topic [**100**], emotion [**113**], situations [**78**], and so on. However, only a few existing studies considered using pre-training based models [**111, 51**]. The basic idea in these previous works is to utilize a parametric vector to represent a condition and then use it in the decoder for conditioned generation. However, the key issue in conditioned dialogue generation is the availability of labeled responses [**117**], and pre-training on unlabeled text or dialogue data does not help much.

Therefore, the motivation of our work is to leverage labeled text (non-dialogue) data that are much easier to collect than labeled dialogue data as supplement. These data can be, for example, texts written by the same person (for a persona condition), within the same topic domain (for a topic condition), etc. The idea is inspired by response style transfer [**56, 60**], which uses a text corpus to learn a style and then transfer the style to dialogue. Based on their success, we assume that the labeled text data can contribute to create better representations of conditions and better utilization of conditions in natural language generation.

In this work, we propose a multi-task learning approach to leverage both labeled dialogue and text data. We use 3 tasks to jointly optimize the same pre-trained Transformer – conditioned dialogue generation task on the labeled dialogue data, conditioned language encoding task and conditioned language generation task on the labeled text data. Our assumption is that the two other tasks can help in our final goal of conditioned dialogue generation: conditioned language generation is the base of conditioned response generation, and conditioned language encoding using bi-directional attention can efficiently encode condition-related expressions and lead to better condition representations. We apply different input representations, self-attention masks, and random mask strategies to differentiate the 3 tasks. Regardless of these differences, the training objectives of these tasks are essentially the same, i.e. masked language modeling, and thus we can mix up 2 types of data / 3 tasks in one training batch, which prevents us from having the catastrophic forgetting problem [**64**].

4.2. Related Work

4.2.1. Conditioned Dialogue Generation

We categorize the related existing works into 3 categories. (1) Response generation conditioned on latent variables, where no extra annotations of dialogues are required [**82, 84, 28, 11, 24, 5**]. (2) Loosely-conditioned response generation, where a label designating the type of the response is required. For example, persona labels [**44**] designate the speaking styles of the responses, and topic labels [**100, 20**] or emotion labels [**47, 113, 71**] specify topic-related or emotion-related vocabularies. These studies usually utilize a parametric vector to encode a label, which is then used in the decoder to guide the generation. (3) Strictly-conditioned response generation, where extra knowledge is required to determine the content of the response, such as a persona profile [**108, 90**], a situation description [**70, 90**], or a wikipedia paragraph [**22, 17**], which are used to ground the response. The ability to strictly-conditioned generation is important, but these dialogues only count for a small fraction of open-domain conversation [**111**]. In many other cases, we are in the situation of loosely-conditioned dialogue. Furthermore, the state-of-the-art strictly-conditioned method [**97**] can be easily added in other models as well [**86, 57**], which simply concatenates the extra knowledge with the dialogue history as the model input.

In this work, we focus on loosely-conditioned response generation [2]. We will show that our approach is robust and can work with different types of labels including those predicted by a classification model, e.g. LDA for topic labels. Therefore, our method is compatible to generation conditioned on latent variables by borrowing power of a classification model. In this work, we do not touch on strictly-conditioned generation. However, this ability can be easily equipped as mentioned.

4.2.2. Response Style Transfer

Style transfer in dialogue aims to learn the style of a text corpus and then incorporate the style in dialogue generation. The transfer is usually between two styles, e.g. rude and polite, or adding a style to general dialogues. To leverage the text corpus, previous work [**56**] jointly trains a Seq2Seq response generator and an extra auto-encoder, and some work [**60**] trains an extra style classifier first to guild the response generator using reinforcement learning.

These works show that text data contain rich information about how to generate a specific type of texts, which inspire us to exploit the labeled text data in conditioned dialogue generation to alleviate the data scarcity issue. Style transfer is usually between two given styles. In contrast, conditioned dialogue generation could work with hundreds of condition

[2]Conditioned generation elsewhere in this chapter refers to loosely-conditioned generation.

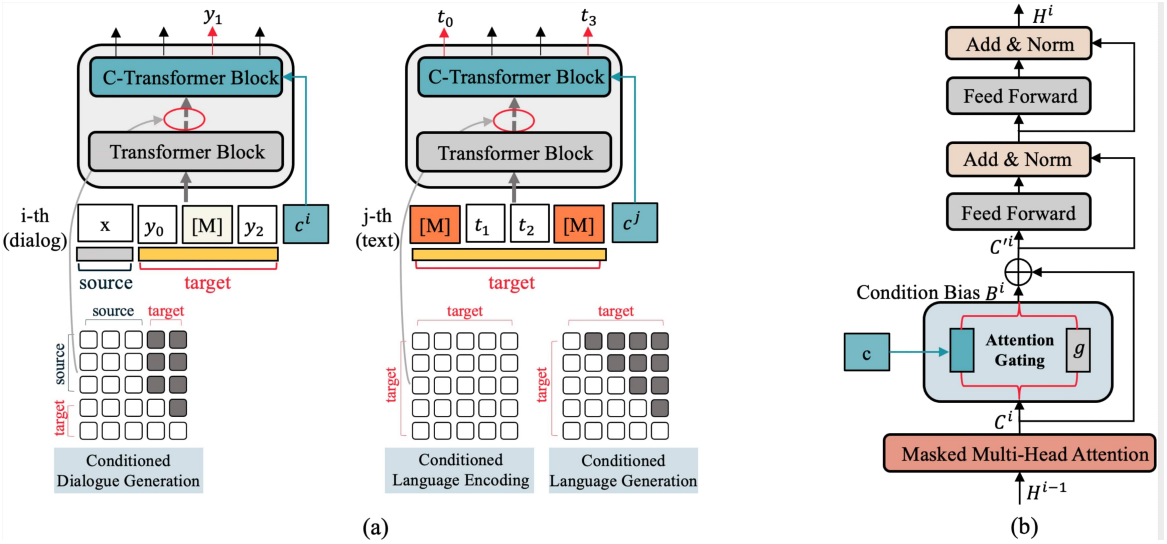

Fig. 4.1. (a) Overview of our multi-task learning approach. Labeled dialogue and text data are mixed, and they are processed using the same pre-trained Transformer with data/task-adaptive input representations, self-attention masks, and random mask strategies. (b) Detailed structures of a condition-aware transformer block, i.e. a C-Transformer Block.

labels simultaneously. As we will show in our experiments, the style transfer methods that utilize additional models, e.g. auto-encoder, to leverage text corpus are unscalable and inefficient for conditioned dialogue. In contrast, our approach that leverages labeled text data without using ad hoc models and makes a tighter integration of labeled text data with labeled dialogue data can more directly impact the conditioned dialogue generation.

4.3. Proposed Method

To efficiently leverage labeled data, first, our approach incorporates all types of data within the same framework, avoiding introducing ad hoc model components which are usually needed in some response style transfer methods in order to leverage extra texts. Second, we propose *TF-IDF based masking* which selects more condition-related tokens to mask, so that the model can exploit the labeled text data more for condition-related expressions rather than the general language features already captured by the pre-trained models. Third, for conditioned generation, we propose a *non-parametric attention-based gating mechanism*, which chooses between generating a general word (necessary for general function words) or a condition-related word at each position. We expect it to be more efficient than a parametric gating. Experimental results show that these approaches all bring improvements.

Our approach is generalizable. In spite of many different labels, a condition essentially specifies some preferences on words, phrases, and sentence structures in the generated responses. Thus, a general approach can be instantiated to a specific case as long as the

corresponding labeled dialogue data are available. We will run experiments with two instantiated models for persona- and topic-related dialogue. Additionally, we will empirically show that our approach is robust and can even work with condition labels predicted by a classification model, e.g. LDA for topic labels.

Dialogue History	Hi Jake, how is your day? [SEP] I am great!
Condition	John
Response	Cool! Do you want to play a video game with me?
Condition	John
	I love playing video games.
Text	My favourite game is GTA5. That's so cool!
	Cool! That's awesome!

Table 4.1. An example of the two types of data that our approach exploits. Here for persona-conditioned dialogue, a condition corresponds to a specific user and encodes some speaking styles.

We assume that we have two types of training data: a labeled dialogue corpus containing (dialogue history, condition, response) samples, and a labeled text corpus consisting of (condition, text) samples. Table 4.1 gives an example of the data. Notice that the "condition" is any categorical label that indicates a type of responses or texts. Our goal is to generate a response y that exhibits the desired characteristics of the type of responses given a dialogue history x and a condition c:

$$y = \arg\max_{y} P(y|x, c) \tag{4.3.1}$$

The Transformer in our work uses bi-directional attention on the source side to encode the dialogue history, and left-to-right attention on the target side to generate the response. Such a transformer can be initialized from BERT[**15**], Roberta[**54**], UniLM [**18**], or the models pre-trained on large-scale unlabeled dialogue data e.g. PLATO [**4**] and Blender [**77**]. In this work, we focus on efficiently leveraging labeled data, i.e. dialogue and text. Figure 4.1 (Left) shows the overview of our approach.

4.3.1. Masked Multi-Head Attention

Section 3.1 introduced the basic components of Transformer. Masked multi-head attention is also applied in our condition-aware transformer block. Our approach jointly optimizes three tasks that apply different self-attention masks as shown in Figure 4.1 (Left). For conditioned dialogue generation task, the self-attention mask allows bi-directional attention on the source side to fully encode dialogue history and left-to-right attention on the target side to generate conditioned responses. For the labeled text data, we randomly choose between conditioned language encoding and conditioned language generation task. The two tasks use

bi-directional attention and left-to-right attention respectively. The language encoding objective, i.e. Masked Language Modeling (MLM), is used in BERT, which has shown stronger ability than the auto-regressive objective used in GPT [**15**]. Therefore, we expect conditioned language encoding is more helpful to learn condition-related expressions (especially with the TF-IDF masking strategy which we will introduce) than the two generation tasks that employ the auto-regressive objective.

4.3.2. Condition-aware Transformer Block

In this subsection, we introduce position-wise condition bias that aims to determine how much condition information should be utilized to bias word generation probability at a position. The core component to calculate the bias is a **non-parametric attention-based gating mechanism** as shown in Figure 4.1 (Right). Other gate mechanisms usually employ parametric linear layers to calculate weights. We assume a self-attention based method (non-parametric) could be more training-efficient, which is important since labeled data are usually limited. We will empirically confirm its effectiveness compared to other gating methods.

Specifically, given a training sample (x, c, y) or (c, text), the condition label c is encoded using two sets of parameters: one parametric vector works as the key $\mathbf{k}^c \in \mathbb{R}^{d_h}$ and another one works as the value $\mathbf{v}^c \in \mathbb{R}^{d_h}$. Additionally, there is a general condition label g with a parametric vector $\mathbf{k}^g$ as its key and a zero vector $\mathbf{v}^g$ as its value. The former corresponds to conditioned generation, while the latter to the general dialogue that generates words only based on dialogue history. At each position, the model determines an attention weight to each choice. More attention to c means that the position is more tuned to the condition. More specifically, for each condition-aware transformer block as shown in Figure 4.1(Right), given $\mathbf{C}^i = [\mathbf{c}^i_1, ..., \mathbf{c}^i_n]$ as queries, the condition biases $\mathbf{B}^i = [\mathbf{b}^i_1, ..., \mathbf{b}^i_n]$ are calculated by:

$$\mathbf{B}^i = \text{softmax}(\frac{\mathbf{C}^i\mathbf{K}_b^T}{\sqrt{d_k}} + \mathbf{M}_b)\mathbf{V}_b \tag{4.3.2}$$

where $\mathbf{K}_b = [\mathbf{k}^c, \mathbf{k}^g]$ and $\mathbf{V}_b = [\mathbf{v}^c, \mathbf{v}^g]$. The calculation is non-parametric. We use the matrix $\mathbf{M}_b \in \mathbb{R}^{n\times 2}$ to prevent adding condition bias to positions on the source side because the condition only influences the target side (the labeled response or text).

4.3.3. Objectives

We jointly optimize three tasks: conditioned dialogue generation on labeled dialogue, conditioned language encoding and conditioned language generation on labeled text. As discussed in Section 3.1, conditioned language encoding is expected to be very helpful to learn condition-related expressions.

A specific self-attention mask is required for each task, while the objectives of three tasks are essentially the same – some tokens of the target side (labeled response or text) are

Dataset	Persona Reddit	Topic dialogue
Source of Labels	Personal ID	LDA
Number of Labels	2000	190
Labeled Texts	3M 500K	3M 500K
dialogue Train	3M 250K	3M 250K
dialogue Valid	80K	80K
dialogue Test	10K	10K

Table 4.2. Key characteristics of the two datasets.

randomly masked, and the final hidden vectors H^L corresponding to the masked tokens are fed into an output softmax over the vocabulary to predict the expected tokens. Therefore, we can mix up 2 types of data (3 different tasks) in one training batch, and the loss is averaged in a batch. This thus prevents us from having the catastrophic forgetting problem [**64**]. This problem is usually observed using a sequential fine-tuning process, i.e. first fine-tuning on labeled texts and then on conditioned dialogue data, which will erase the effect of the previous steps of training.

When using labeled dialogue data, we want the model to learn to generate conditioned but more importantly coherent responses. Thus, we uniformly sample the tokens on the target side to mask. Differently, when exploiting labeled text data, we only want the model to generate condition-related expressions. Therefore, we introduce **TF-IDF Based Masking** for the labeled text data to speed up the learning process – we sample tokens to mask according to their TF-IDF values counted on the entire corpus. We will empirically show its effectiveness.

4.4. Experiments

4.4.1. Datasets

We use two labeled dialogue datasets, and we created two smaller training sets (500K labeled texts and 250K labeled dialogues), which are summarized in Table 4.2. We anticipate that when labeled dialogue data are limited, the benefit of leveraging labeled text data will be larger.

Persona Reddit We filtered the Reddit data from 2015 to 2019 that is provided by a third party [3]. Reddit data is a natural source of dialogue with multiple users – a post may have multiple comments by different users. Following previous work [**44**], we consider each user as a distinct persona. We extract (post, user, comment) tuples, where "user" is the label of the user who makes the "comment". We further filtered the data based on sentence length and users: sentences with more than 30 words or less than 4 words are removed, and

[3] `https://files.pushshift.io/reddit/`

we only keep comments from the 2000 most active users so that we can collect enough data for each user. As a result, each user has 1291 samples (comments) on average. To build the labeled text corpus, we collect extra posts or comments on Reddit from the same user that have no overlap with the dialogue data – these extra texts are intended to reflect the general writing style of the user.

Topic-related Dialogue previous work [**20**] provides a high-quality 3-turns conversational dataset for topic aware response generation [4]. Along with each (history, target) pair, there is a topic label and dozens of topic words that are predicted by LDA model. The dataset contains 9.2M samples, from which we sample 3M (history, topic, target) tuples as the labeled dialogue corpus. To construct the labeled text data, we sample other 3M tuples and only keep their (topic, target) parts. Note that the topic labels are generated by LDA, and thus it is difficult to obtain the labeled text data from other sources.

4.4.2. Baselines

We choose two strong baselines specifically designed for personalized response generation and two others for topic-aware generation. Additionally, we choose some state-of-the-art pre-trained Transformers.

Speaker Model[**44**] a Seq2Seq model using four LSTM layers. Given a user label, the decoder transforms it into a user embedding and use it to generate a personalized response.

MT-Speaker an approach jointly trains a Speaker Model and a conditioned auto-encoder with shared decoder parameters, which is adapted from a style transfer approach [**56**]. This approach also leverages the labeled text data.

TA-Seq2Seq [**100**] and **THRED** [**20**] these models utilize topic words instead of topic labels predicted by the LDA model. TA-Seq2Seq leverages the topic information by a joint attention mechanism and a biased generation probability. THRED is built based on HRED and incorporates topic words via a hierarchical joint attention mechanism.

C-Trans-ED [**111**] an encoder-decoder transformer framework initialized with GPT parameters. The decoder dynamically merges features from the dialogue history and the condition. This model is based on the code of ConvAI2 champion [**16**].

C-Trans-Dec a decoder-only transformer initialized with GPT-2 parameters, adapted from previous work[**97**]. We add a condition embedding to the input representation to enable conditioned generation.

BERT fine-tuning the pre-trained model [**15**] on the dialogue datasets. The encoder and decoder share the parameters. When encoding, the model uses bi-directional attention. When decoding, it uses left-to-right attention.

[4] `https://github.com/nouhadziri/THRED`

Model	Parameters	Runtime(min/M)
Sp-Model	80M	25
MT-Speaker	90M	40
TA-Seq2Seq	155M	150
THRED	174M	135
C-Trans-ED	120M	180
C-Trans-Dec	126M	290
BERT	110M	140
Ours	113M	145

Table 4.3. The number of parameters of each tested approach and the average runtime (minutes) for every million training samples.

4.4.3. Implementation Details

We implement the speaker model and MT-Speaker model based on OpenNMT [5]. Other models are directly taken from the available open-source code. Hyper-parameters are set following the original papers. Since our baselines utilize GPT or BERT, we use BERT (base, uncased) to initialize our model for fair comparison. It is however possible to build our model upon more powerful pre-trained dialogue models such as Blender [**77**] or PLATO[**5**]. We do hyper-parameter search based on perplexity [6] on the validation set for: the number of condition-aware transformer blocks in $\{2, 6, 12\}$, the mix-up rate of labeled dialogues and texts in {3:1, 1:1}, and whether using conditioned language encoding task. We report experimental results with 2, 3:1, and using conditioned language encoding respectively. The warm-up proportion is set to 0.1. 25% tokens of the target side are randomly masked. During decoding the beam size is 10, and we prevent duplicated bigrams. We fine-tune all the parameters end-to-end on two P100 GPUs. Generally, we used early stop according to performance observed on validation set to prevent over-fitting. For large-scale datasets, performances of models usually stop to increase a lot after the 4-th epoch. With in total 6M training samples, each epoch takes twelve hours. The fine-tuning model only has $(2C+1)\times d_h$ additional parameters, where C is the number of different condition labels.

In Table 4.3, the average runtime is tested using a 1080Ti GPU device, and the batch size is set to take all of the GPU memories. TA-Seq2Seq and THRED are implemented in TensorFlow. Other models are implemented in PyTorch. Notice that the runtime will be influenced by code implementation in additional to model structure. When experimenting with the small-scale Persona Reddit dataset, we decrease the number of parameters of Sp-Model and MT-Speaker models to 48M and 52M respectively in order to avoid over-fitting.

[5] `http://opennmt.net/`

[6] Perplexity evaluates how likely the model generates the ground-truth responses. So we use it as the proxy for model fitness.

Hyper-parameters	Value
C-Tranformer layers	2
mask probability	0.25
max length	80
batch size	160
learning rate	3e-5
warmup proportion	0.1
label smoothing	0
weight decay	0.01
dropout probability	0.1

Table 4.4. Hyper-parameters for our fine-tuning approach. There are in total 6M data. Thus, we use a large batch size.

C-Trans-ED loads the pre-training results of GPT. In the original paper, they pre-trained by themselves using a Chinese corpus, which cannot be used in our experiments.

4.4.4. Evaluation

Automatic Metrics We choose some widely used metrics in the literature [7]: **BLEU** [**61**] with n=1,2,3; **ROUGE-L** – longest common subsequence based statistics; **CIDEr** [**92**] utilizing TF-IDF weighting for each n-gram; and **Distinct** [**43**] indicating the proportion of unique n-grams (n=1,2) in the entire set of generated responses to evaluate response diversity. Two-sided t-test is used for statistical significance test.

Response Appropriateness Furthermore, we conduct manual evaluation on the best models according to the automatic metrics. We only manually evaluate the model performance on large-scale datasets[8]. We ask human evaluators to rate a response in $\{0, 1, 2\}$. A score of 0 means that the response might have flaw in fluency and logic or be incoherent. Special cases are for example completely coping from the dialogue history as the output, and a bland response such as "I don't know what you mean". A score of 1 represents a coherent but generic response. 2 represents a coherent and informative response. We also do a pairwise evaluation to compare two models and indicate which one is better. The evaluation is based on 200 random samples. Each generated response is rated by three annotators. The inter-rater annotation agreement in Cohen's kappa [**14**] is 0.441 on average, which indicates moderate agreement.

Condition Consistency We observe that automatic metrics fail to evaluate condition consistency since BERT that does not consider conditions outperforms C-Trans-ED and C-Trans-Dec. Thus, we perform manual evaluation on the condition consistency. A generated

[7]We use an open-source evaluation tool: `https://github.com/Maluuba/nlg-eval`

[8]We did not manually evaluate the results with small datasets due to its high cost. However, we expect even larger difference when small data are used for training, as indicated by the automatic metrics.

Model	BLEU-1	BLEU-2	BLEU-3	ROUGE-L	CIDEr	Dist-1	Dist-2	avgLen
Sp-Model	10.539 (**)	3.152 (**)	1.396 (**)	0.116 (**)	0.056 (**)	0.012 (**)	0.044 (**)	12.6
MT-Speaker	10.970 (**)	3.488 (**)	1.540 (**)	0.118 (**)	0.059 (**)	0.009 (**)	0.034 (**)	12.7
C-Trans-ED	13.548 (*)	3.881 (**)	1.529 (**)	0.113 (**)	0.045 (**)	0.005 (**)	0.025 (**)	18.7
C-Trans-Dec	12.964 (**)	4.182 (**)	1.781 (**)	0.117 (**)	0.060 (**)	0.023 (**)	0.097 (**)	16.7
BERT	12.928 (*)	4.405 (/)	1.764 (**)	0.119 (**)	0.062 (**)	0.014 (**)	0.052 (**)	26.1
Ours	**14.052**	**4.891**	2.149	**0.122**	0.070	**0.024**	0.098	23.3
Two-Step FT	13.714 (/)	4.870 (/)	**2.160** (/)	**0.122** (/)	**0.071** (/)	0.023 (/)	0.102 (*)	25.0
w/o ctext	13.015 (*)	4.563 (/)	1.956 (/)	0.113 (**)	0.061 (**)	0.023 (/)	**0.106** (*)	25.7
w/o tfidf	13.581 (*)	4.705 (/)	2.000 (/)	0.118 (**)	0.070 (/)	0.023 (/)	0.095 (*)	24.0
Sp-Model	10.467 (**)	3.039 (**)	1.239 (**)	0.116 (**)	0.049 (**)	0.007 (**)	0.027 (**)	12.3
MT-Speaker	10.286 (**)	2.932 (**)	1.174 (**)	0.114 (**)	0.047 (**)	0.007 (**)	0.030 (**)	12.3
C-Trans-ED	10.968 (**)	3.247 (**)	1.295 (**)	0.106 (**)	0.040 (**)	0.001 (**)	0.006 (**)	14.7
C-Trans-Dec	11.263 (**)	3.390 (**)	1.274 (**)	0.106 (**)	0.043 (**)	0.020 (**)	0.075 (**)	16.2
BERT	12.766 (*)	4.195 (*)	1.805 (*)	0.118 (/)	0.063 (*)	0.022 (/)	0.071 (**)	15.3
Ours	**13.517**	**4.517**	**1.988**	**0.119**	**0.068**	0.021	0.066	16.4
Two-Step FT	10.125 (**)	3.295 (**)	1.388 (**)	0.111 (**)	0.052 (**)	0.015 (**)	0.043 (**)	12.7
w/o ctext	11.776 (**)	3.821 (**)	1.631 (**)	0.115 (**)	0.059 (**)	0.020 (*)	0.062 (**)	14.4
w/o tfidf	13.475 (/)	4.409 (/)	1.853 (/)	0.118 (/)	0.064 (*)	**0.023** (/)	**0.078** (*)	16.7

Table 4.5. Evaluation results on large-scale (upper half) and small-scale (lower half) Persona Reddit. Two-Step FT means using our model architecture but applying sequential fine-tuning. w/o ctext is without leveraging conditioned text data. w/o tf-idf means without applying TF-IDF based masking. * ($p < 0.05$) or ** ($p < 0.01$) show statistically significant differences with our model by two-sided t-test.

response is rated in $\{0, 1, 2\}$. The scores 0, 1 and 2 mean respectively that the response is inconsistent to the condition, somehow related, and consistent. However, if the response has flaw in fluency or logic, it will get a score of 0. For Topic Dialogue, it is easy to measure whether a generated response is in the topic. However, for persona consistency, it is difficult for a human evaluator to know the speaking style of each user. Thus, before evaluation we first automatically determine those frequently used words by a user in responses and show them to the annotators to help their evaluations.

4.4.5. Analysis

Table 4.5 and 4.6 gives automatic evaluation results, and Table 4.7 gives human evaluation results. The results can be summarized as follow:

BERT vs. Trans-ED & Trans-Dec C-Trans-Dec has a clear advantage over C-Trans-ED in almost all automatic metrics, which can also be observed in their generated responses. Fine-tuning BERT without considering conditions outperforms C-Trans-Dec on most similarity metrics such as BLEU. We explain this by the fact that bi-directional attention could enable a model to better encode dialogue history, and thus to generate responses more similar to the ground truth. The ablation model using w/o ctext is fine-tuning C-BERT (with our condition-aware transformer blocks) on labeled dialogue data. The performance of w/o ctext is similar to C-Trans-Dec's, with a slight advantage in condition consistency and small

Model	BLEU-1	BLEU-2	BLEU-3	ROUGE-L	CIDEr	Dist-1	Dist-2	avgLen
TA-Seq2Seq	10.197 (**)	3.307 (**)	1.602 (**)	0.121 (**)	0.098 (**)	0.016 (**)	0.051 (**)	9.7
THRED	9.061 (**)	3.035 (**)	1.468 (**)	0.118 (**)	0.098 (**)	0.015 (**)	0.048 (**)	8.8
C-Trans-ED	13.990 (**)	5.359 (**)	2.689 (**)	0.131 (**)	0.147 (**)	0.055 (**)	0.222 (**)	12.5
C-Trans-Dec	14.544 (**)	5.475 (**)	2.669 (**)	0.136 (**)	0.154 (**)	0.046 (**)	0.177 (**)	13.2
BERT	15.287 (/)	6.243 (/)	3.283 (/)	0.141 (/)	0.168 (**)	0.057 (**)	0.227 (**)	12.5
Ours	15.639	**6.484**	**3.455**	0.140	0.185	0.060	0.243	13.0
Two-Step FT	**15.926** (/)	6.431 (/)	3.376 (/)	**0.143** (/)	0.185 (/)	0.059 (*)	0.239 (*)	13.1
w/o ctext	15.491 (*)	6.397 (/)	3.399 (/)	0.142 (/)	**0.190** (*)	**0.063** (*)	**0.262** (**)	12.8
w/o tfidf	15.393 (/)	6.302 (/)	3.351 (/)	0.139 (/)	0.185 (/)	0.059 (**)	0.230 (**)	13.1
C-Trans-ED	13.874 (**)	5.145 (**)	2.503 (*)	0.124 (**)	0.124 (**)	0.039 (**)	0.150 (**)	13.1
C-Trans-Dec	**14.899** (/)	5.648 (/)	2.690 (/)	0.133 (**)	0.150 (/)	0.043 (*)	0.176 (*)	15.2
BERT	14.457 (/)	5.583 (/)	2.802 (/)	0.135 (**)	0.136 (**)	0.037 (**)	0.133 (**)	12.4
Ours	14.587	**5.747**	**2.894**	**0.139**	**0.152**	**0.050**	**0.186**	12.0
Two-Step FT	13.941 (**)	5.463 (/)	2.765 (/)	0.136 (*)	0.140 (**)	0.045 (**)	0.169 (**)	11.7
w/o ctext	13.211 (**)	5.179 (**)	2.655 (/)	0.137 (/)	0.142 (**)	0.046 (**)	0.163 (**)	10.8
w/o tfidf	13.964 (**)	5.485 (**)	2.809 (/)	0.135 (**)	0.145 (*)	0.048 (*)	0.178 (**)	11.8

Table 4.6. Evaluation results on large-scale and small-scale Topic Dialogue. Topic labels are predicted by LDA.

Model	Persona				Topic			
	Appropriateness		Consistency		Appropriateness		Consistency	
	Score	Pair-wise	Score	Pair-wise	Score	Pair-wise	Score	Pair-wise
C-Trans-Dec	0.96	(28%, 39%)	0.85	(20%, 39%)	0.77	(26%, 34%)	0.71	(21%, 31%)
BERT	0.77	(11%, 40%)	0.78	(22%, 43%)	0.55	(17%, 40%)	0.46	(16%, 40%)
Ours	**1.15**	-	**1.24**		**0.83**	-	**0.80**	-
w/o ctext	0.91	(26%, 39%)	0.90	(23%, 38%)	0.73	(27%, 35%)	0.72	(23%, 30%)

Table 4.7. Human evaluation of generated responses on appropriateness and condition consistency. Pair-wise comparisons show the wining percentages of (baseline, ours).

disadvantage in response appropriateness. These results show that our approach is built upon a strong base model. As mentioned, other pre-trained models can also be used.

With Condition When large Persona Dialogue is available, w/o ctext (i.e. C-BERT) outperforms BERT in almost all automatic metrics. However, we observe that when only small-scale labeled dialogue data are available, all three conditioned models perform worse than BERT. This shows that the model cannot learn the condition-related features well from the limited labeled dialogue data. Thus, it is important to leverage the labeled texts that are easier to collect, and the results on small-scale Persona Reddit show that our multi-task learning approach significantly outperforms BERT on similarity metrics such as BLEU and CIDEr.

For Topic Dialogue, the labels are given by LDA model. LDA is an unsupervised method and the predicted condition labels can be very noisy. Nevertheless, similarly, with large data C-BERT outperforms BERT in all metrics, but when only small-scale labeled dialogue data are available, C-BERT performs worse than BERT in terms of BLEU. The result again shows

dialogue His.	ca n't wait to see the team at full strength when watt and clowney are back .
Gold Resp.	i 've been waiting a long time to see watt and clowney healthy together
Sp-Model	i do n't think you know what the word means (0)
MT-Speaker	i do n't think the texans will be able to handle the ball in the first half (2) of the season
C-Trans-ED	i do n't know why you 're getting downvoted, i 'm just saying that (0) you 're wrong .
C-Trans-Dec	i 'm not sure if i 'd be surprised if they both get hurt . (1)
BERT	i ca n't wait to see watt and clowney at full strength when watt is back . (0)
Ours	i do n't think it 's going to happen , but i 'd love to see watt and clowney back . (2)
w/o ctext	i do n't think it 's going to happen , but i 'd love to see watt and clowney on the same team (2)
w/o tfidf	i do n't think it 's going to happen , but i 'd love to see watt back . (2)
Sp-Model	i do n't think he is going to be a good qb . (1)
MT-Speaker	i 'm not sure if this is a good thing . (2)
C-Trans-ED	i 'm not sure what you 're talking about . i do n't know what to say . (0)
C-Trans-Dec	i 'm not sure i 'd be surprised if they 're both back (2)
BERT	i do n't think they 'll be able to do that . (2)
Ours	i 'd be so happy to see watt and clowney back . (2)
Two-Step FT	i do n't think that 's what i 'm talking about . (0)
w/o ctext	i do n't think they 'll be able to do that . (2)
w/o tfidf	i 'm so excited to see watt on the team . (2)

Table 4.8. Responses generated by baselines and our model trained on the large-scale and small-scale Persona Reddit.

the importance of exploiting labeled texts, and our approach is the best on small-scale Topic Dialogue.

Leveraging Labeled Texts In general, our approach significantly outperforms all baselines and w/o ctext that do not exploit labeled text data, either with large-scale or small-scale data. With small-scale data, our approach outperforms BERT while w/o ctext itself cannot achieve this, which shows that conditioned dialogue generation can be helped by extra labeled text data. On Topic Dialogue, with such noisy labels, our model leveraging the labeled texts still produces the best performance, which confirms the robustness of our multi-task learning approach to work with different types of labels. The human evaluation on appropriateness and condition consistency further confirms the effectiveness of our approach.

Not all methods utilizing extra labeled text can obtain such performance improvement as we did. MT-Speaker that employs an extra auto-encoder does not gain much improvement over Sp-Model. This result shows that using additional model components to leverage labeled texts is inefficient for conditioned dialogue generation. Furthermore, Two-Step FT that first fine-tuning on labeled texts and then on labeled dialogue data does not always produce good performance. It achieves comparable performance to our approach on large-scale datasets, but on small-scale datasets it can even perform worse than w/o ctext (Table 4.5). This result shows that the size of labeled text data is a considerable factor. With small-scale labeled text

dialogue His.	what is your favorite food ? [SEP] ice cream .
Gold Resp.	what flavor ?
TA-Seq2Seq	what kind of ice cream ? (2)
THRED	what kind of ice cream ? (2)
C-Trans-ED	ice cream is the best food i have ever eaten (2)
C-Trans-Dec	i 'm not sure if you 're joking but i 'm not sure if you 're serious . (0)
BERT	i do n't like ice cream . (2)
Ours	i do n't like ice cream . it 's coarse and irritating and it gets everywhere . (2)
w/o ctext	i 've never had ice cream . what 's it taste like ? (2)
w/o tfidf	what kind of ice cream do you like ? (2)
C-Trans-ED	ice cream is the best food i 've ever seem . (2)
C-Trans-Dec	i 've never had ice cream . (2)
BERT	i 've never had ice cream . (2)
Ours	i do n ' t like ice cream . (2)
Two-Step FT	i like ice cream , but i do n ' t like it . (0)
w/o ctext	i 've never had ice cream , but it 's so good . (2)
w/o tfidf	i ' ve never had ice cream . (2)

Table 4.9. Responses generated by baselines and our model trained on the large-scale and small-scale Topic Dialogue. We use a special token, [SEP], to concatenate successive dialogue turns.

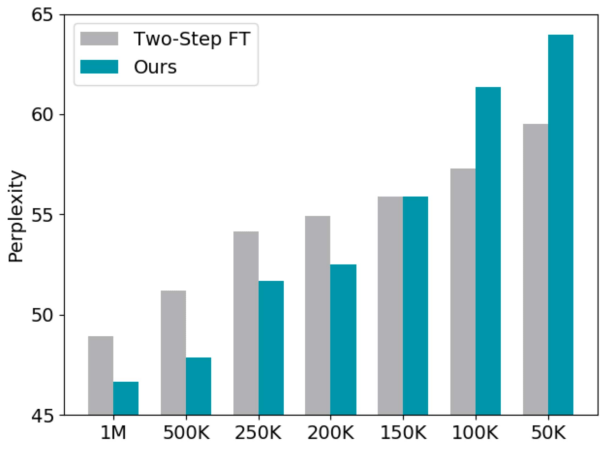

Fig. 4.2. Perplexity of the model with sequential fine-tuning and our approach given 1M labeled text data and different size of labeled dialogue data.

data, it is better to avoid sequential fine-tuning that has the catastrophic forgetting problem [**64**]. Additionally, we investigate how the ratio of the size of labeled text data to the size of dialogue data influence model performance. We use perplexity to measure the performance of a model: a better model is the one leading to lower perplexity. As shown in Figure 4.2, given 1M labeled text data, when the ratio is less than 6.7, our approach performs better than Two-Step FT. However, when labeled text corpus is much larger than dialogue corpus, sequential fine-tuning is better.

Model	BLEU-1	BLEU-2	Dist-2
Single Gate	13.880 (*)	4.853 (/)	0.090 (**)
Double Gates	13.988 (*)	4.889 (/)	0.094 (*)
Attn. Routing	**14.052**	**4.891**	**0.098**
Single Gate	11.703 (**)	3.891 (**)	0.090 (**)
Double Gates	11.336 (**)	3.698 (**)	**0.091** (**)
Attn. Gating	**13.517**	**4.517**	0.066

Table 4.10. Comparison of gating mechanisms on large-scale and small-scale Persona Reddit.

TF-IDF Masking and Attention Gating We assumed that the general language features have already been captured by the pre-trained models. Thus, to better utilize labeled text data, we mask more condition-related words using TF-IDF based masking. Our ablation study confirms that TF-IDF masking brings improvement in almost all automatic metrics although the improvement might not always be statistically significant.

Our attention gating is a non-parametric gating mechanism to fuse the condition into the decoder. We expected it to be efficient, which is particularly important when labeled data are limited. Here, we compare it with two common parametric gating mechanisms: 1) setting a single gate on $\mathbf{C}^i$ to get a weight; 2) setting gates on both $\mathbf{C}^i$ and $\mathbf{v}^c$ to get two weights. Then, we combine the weighted $\mathbf{C}^i$ and $\mathbf{v}^c$ to get $\mathbf{C}'^i$ as in our attention gating. Experimental results in Table 4.10 confirm that our method is more efficient. When only small-scale labeled data are available, the model with attention gating generates responses that are significantly more similar to the ground-truth.

4.5. Conclusion

In this chapter, we examined the data scarcity issue of conditioned dialogue generation. Pre-training on unlabeled text or dialogue data is not helpful to conditioned generation. Thus, we exploited labeled text data that are easier to collect than labeled dialogues. We expected these data can contribute to better representations of conditions and better use the conditions in natural language generation, which complement what is lacking in the pre-trained models.

To leverage these two types of data, we proposed a simple and efficient multi-task learning approach. Three tasks are considered: conditioned dialogue generation task on the labeled dialogue data, conditioned language encoding task and conditioned language generation task on the labeled text data. We conducted experiments under persona and topic conditions. Experimental results show that our approach outperforms the state-of-the-art models by leveraging labeled texts, and it also obtains larger improvement in performance comparing to the previous methods leveraging text data.

Chapter 5

Adapter based on Pre-trained Model for Dialogue Skill Learning

Dialogue models pre-trained on large unlabeled conversation data (e.g. Reddit and Twitter) [**109, 1, 77**] have already shown excellent performance in generating coherent and fluent responses. Nevertheless, a recent study [**77**] shows that large additional improvements can be obtained by fine-tuning a pre-trained dialogue model on data that emphasizes desirable conversational skills. By conversation skills, we mean the ability of the system to use some information and knowledge related to the conversation context. The conditioned dialogues we presented in the previous chapter can be viewed as skills, i.e. the skills to exploit conditioned data. More generally, dialogue skills refer to the general, domain-independent functionality of open-domain dialogue systems to generate responses that are consistent with some conditions, e.g. a pre-defined persona profile.

It is important to underline the difference between the general dialogue skill that exploits persona-related information and the the ability of a system to generate persona-related responses. To generate persona-related dialogue in the latter case, one can train (fine-tune) a model for the persona specifically. However, for a new persona, one has to retrain the whole model once again. No common "skill" is generated from the first persona, and transferable to the second one. On the other hand, the dialogue skill learned from a person is more general, and can be transferred to a new persona. A more concrete example of skill is the ability of a system to exploit the information about the profession of the persona, independently of the specific profession and persona. Once the skill is acquired, it can be applied to another persona with a different profession. In other words, dialogue skills correspond to the "know-how" while a specific conditioned dialogue corresponds to "know".

Dialogue skills cannot be learned by reconstructing ground-truth responses, which will mix up skills with domain-related expressions. These latter are not transferable to other domains. In this chapter, we propose a novel training approach aiming at extracting the transferable dialogue skills underlying task(s). The approach leverages an auxiliary training

objective, multi-task learning as well as small adapters to constrain the skill model to focus on the general skill across domains. Experimental results show that our approach can effectively learn dialogue skills in a multi-domain dialogue context.

5.1. Definition of Dialogue Skill

Dialogue skill is defined as the general ability of a dialogue agent to exploit some knowledge or information [**86, 77**]. For example, the agent should know how to use the persona information in dialogue [**108**], or how to exploit knowledge [**17**]. This is in contrast to domain-specific features such as specific expressions in a domain. A difference between them is that the former is general and more abstract, while the latter is more specific to a domain. We can also use the term "know-how" to refer to the former and "know" to the latter.

Several small single-skill datasets have been created for communicating with a persona profile [**108**], displaying empathy [**71**], and answering questions by utilizing knowledge resources [**17**]. In these studies, a dialogue skill model is viewed as what can help to generate responses that fits the task context such as persona profile or knowledge. As a matter of fact, attempts have been made to create multi-skill dialogue systems [**86, 77**] by leveraging multi-task learning [**9**] on several single-skill datasets. However, these attempts have not been successful in the sense that the performance of such a system is worse than the one fine-tuned on each dataset, showing what is learned from a dataset is hard to be transferred to another. A key issue is about the training objective: These models have been trained using the standard Negative Log Likelihood Loss (NLLLoss) trying to reconstruct the ground-truth responses. However, at the same time, the model would also capture domain-specific expressions, making it hard to be transferred to another domain.

We believe that it is critical to design an auxiliary task in order to extract a specific skill. Therefore, we propose a novel training approach to capture the dialogue skill underlying task(s). To this end, first, we require the model to explicitly predict whether a condition is relevant to dialogue history so that the model should use it in generation. This auxiliary task can also be applied for training other skills such as knowledge skill [**17**] that also requires to recognize useful information before generation. Next, we train the model by conditioned response generation task based on relevant conditions instead of all conditions. To implement this, however, we will have to take into account the possible noise in the determined condition(s). Therefore, we propose a more robust training process in such a noisy situation.

In this chapter, we focus on learning domain-independent dialogue skills that can be transferred more easily. In order to do this, we should have appropriate datasets for the training and evaluation, which are unfortunately lacking. Therefore, we first construct a dataset, named PersonaSkillTalk, based on two public datasets: PersonaChat [**108**] and

LIGHT [**90**]. Both of them are conditioned on persona profiles, thus they require using similar dialogue skills relating to persona-conditioned dialogue. However, there is a clear difference between them: PersonaChat is about daily life and LIGHT is about adventure game. Using this dataset, we will be able to test if a dialogue skill related to persona is learned: If a model learns the skill on the data in one domain, transferring it to another domain (requiring the same skill) would yield better performance.

Through experiments on PersonaSkillTalk, we will confirm that using NLLLoss only makes the model capture everything that is specific to the dataset, including the skill and domain-specific features. We will also introduce a promising avenue toward learning transferable dialogue skills.

5.2. Related Works

5.2.1. Dialogue Skill Modelling

Previous works [**86, 77**] view a dialogue skill as the ability to generate responses by taking into account the task context such as persona profile [**108**], situation [**71**], knowledge [**17**] or image [**85**]. To build a multi-skill dialogue model, [**86**] and [**77**] implement multi-task learning [**9**] on multiple single-skill datasets. However, the network captures only the minimal common structure underlying all the datasets, and thus multi-task learning has shown to perform worse than fine-tuning on each dataset. To remedy the problem, [**87**] propose to train a dialogue manager (a classifier) to switch among skills so as to combine multiple single-skill dialogue models trained in advance. [**57**] reduces the parameters of the multi-skill dialogue model by replacing a large single-skill dialogue model with a small adapter [**33**]. These adapters for different skills are then employed upon a fixed pre-trained dialogue model.

These previous works assume that they have trained a single-skill model, and focus on how to combine different skills. However, our experiments show that all these studies using NLLLoss only and trained on a single dataset cannot learn a transferable dialogue skill. Thus, they will not be able to combine skills of different domains or to transfer skills to other domains. Their approaches only work if all single-skill models are in the target domain. In this work, we investigate how to learn a transferable dialogue skill, and we will validate the effectiveness of our approach on persona-conditioned datasets.

5.2.2. Dialogue Training Objective

The standard approach to train dialogue response generator is minimizing NLLLoss given ground-truth responses. Some works have applied other training objectives to improve response specificity [**79**] and dialogue coherence [**95, 102, 46**]. However, previous works on

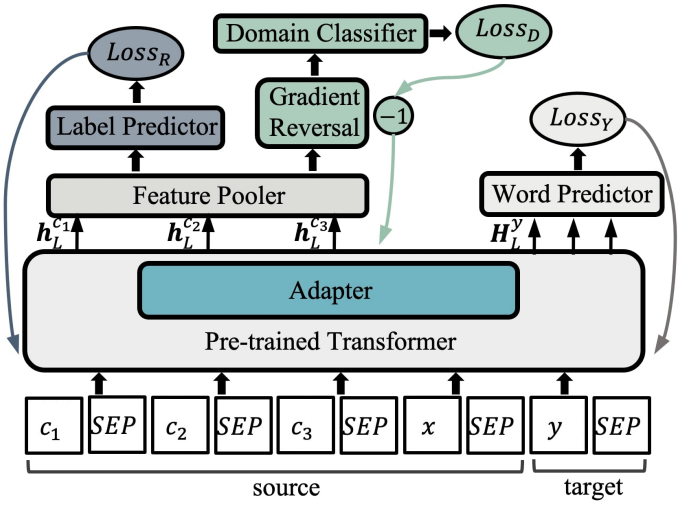

Fig. 5.1. Architecture and objectives for dialogue skill training. In addition to ground-truth response reconstruction ($\mathcal{L}_Y$), there are relevant condition recognition ($\mathcal{L}_R$) and domain adversarial training ($\mathcal{L}_D$). The Predictor, Classifier and Pooler are a linear layer with an activation function.

dialogue skills are trained using NLLLoss only [**86, 77, 87, 57**]. Since this training objective requires the model to fully reconstruct the ground-truth responses, the model will also learn many domain-specific expressions, especially when training the model on a single dataset. Our experiments will examine this problem. Therefore, in this work, we propose to use an auxiliary loss to help capture the dialogue skill underlying dataset(s), and we will empirically show its usefulness.

5.2.3. Adapter

An adapter is a set of trainable parameters that steer a fixed base model to a down-stream task. Only the adapter is fine-tuned in the training process. Adapter has been shown to yield parameter-efficient tuning for NLP [**33**]. This ability is desirable to build a multi-skill dialogue model [**57**]. Many adapter variants have been applied to diverse tasks including visual domain learning [**73**], language adaptation [**6, 63**], and knowledge infusion [**93**]. What we contribute in this work is that we find that adapter helps to learn a transferable dialogue skill since the small model capacity prevents it from learning diverse domain-specific features.

5.3. Methodology

A single-skill dialogue dataset is denoted as $\mathcal{D} = \{(\mathbf{C}^{(i)}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$, where $\mathbf{x}^{(i)}$ is the dialogue history, $\mathbf{y}^{(i)}$ is the ground-truth response, and $\mathbf{C}^{(i)} = \{\mathbf{c}_1^{(i)}, ..., \mathbf{c}_k^{(i)}, ...\}$ are conditions related to the skill, e.g. persona profiles for persona skill. Given $\mathbf{c}_k^{(i)}$ and $\mathbf{y}^{(i)}$, we assume that

it is known if the condition is utilized in the response (see section 5.4.1 for data annotation). Based on it, $\mathbf{r}_k^{(i)} \in \{0, 1\}$ indicates whether $\mathbf{c}_k^{(i)}$ is relevant to $\mathbf{x}^{(i)}$.

We apply a pre-trained Transformer-AR [**86, 106**] as the base dialogue model, which is a decoder-only framework that uses bi-directional attention on the source side and left-to-right attention on the target side. This framework enables the application of a classification task on the source side, and thus we will introduce relevant condition recognition as the auxiliary training objective. Figure 5.1 illustrates the components that we will compare in experiments.

5.3.1. Training Objective

We train a model to learn a dialogue skill by maximizing two probabilities, namely $\sum_k P(\mathbf{r}_k^{(i)}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)})$, and $P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{C}_R^{(i)})$, where $\mathbf{C}_R^{(i)}$ is the `relevant` conditions. In contrast, previous works employ NLLLoss to train the model, i.e. maximizing $P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)})$, expecting to implicitly optimize relevant condition recognition and conditioned response generation. However, we observe in our experiments that the method also learns many domain-specific features (/expressions).

Relevant Condition Recognition. To capture the underlying dialogue skill, we ask the model to explicitly predict whether a condition is relevant to dialogue history so that the model should use it in generation. Then, we minimize:

$$\mathcal{L}_{\mathrm{R}}^{(i)} = -\sum_{k=1}^{|\mathbf{C}^{(i)}|} \log P_k(\mathbf{r}_k^{(i)}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)}), \tag{5.3.1}$$

We utilize $\mathbf{h}_L^{c_k}$ to calculate the probability P_k, which is the output of the transformer at $\mathbf{c}_k^{(i)}$ [SEP] position. This position only attends to the tokens within $\mathbf{c}_k^{(i)}$ and thus aggregates the information in this range. The representation at the [SEP] position can then be considered as the representation of $\mathbf{c}_k^{(i)}$. In contrast, the general attention on the source side is bi-directional, which ensures that the probability is calculated based on $\mathbf{x}^{(i)}$ and $\mathbf{C}^{(i)}$.

This auxiliary task can also be applied for knowledge-conditioned dialogue generation [**17**] that also requires to recognize useful information first. However, this task is inapplicable to image-conditioned [**85**] or situation-conditioned generation [**71**]. In these cases, other auxiliary tasks that help to learn the underlying dialogue skill need to be explored.

Conditioned Response Generation. Since we have separated the relevance recognition process, for conditioned generation we use ground-truth $\mathbf{C}_R^{(i)}$. Specifically, we minimize the negative loss likelihood:

$$\mathcal{L}_{\mathrm{Y}}^{(i)} = -\sum_{t=1}^{|y^{(i)}|} \log P_t(y_t^{(i)}|\mathbf{x}^{(i)}, \mathbf{C}_{\hat{R}}^{(i)}, y_{<t}^{(i)}), \tag{5.3.2}$$

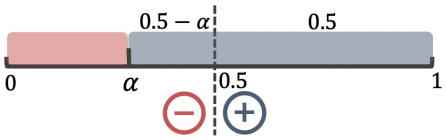

Fig. 5.2. In inference, only when $P(\texttt{relevant}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)}) < \alpha$ will the generation not attend to this condition.

According to the formulation, this objective inevitably forces the model to learn the tokens/expressions closely related to the dataset while learning conditioned generation.
Training under Noisy Conditions (Denoising). We observe that using ground-truth $\mathbf{C}_R^{(i)}$ for training is inappropriate since in inference the predicted relevant conditions could be inaccurate and contain noises. Therefore, we design a training process that simulates the same noisy situations. In addition to the ground-truth condition, we randomly sample some irrelevant conditions for training[1]:

$$\mathbf{C}_{\hat{R}}^{(i)} = \mathbf{C}_R^{(i)} + \texttt{sample}(\mathbf{C}_{IR}^{(i)},\ \texttt{p} = \beta), \tag{5.3.3}$$

where each irrelevant condition is uniformly sampled with the probability β. In inference, only when $P_k(\texttt{relevant}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)}) < \alpha$, the condition will be viewed as irrelevant, and the generation will not attend to it. The sampling probability β is calculated based on α to ensure that the noises introduced in the training process is consistent with inference. In inference, only when $P_k(\texttt{relevant}|\mathbf{x}^{(i)}, \mathbf{C}^{(i)}) < \alpha$, the condition will be viewed as irrelevant, and the generation will not attend to it. It can be viewed as: $\frac{\texttt{false}}{\texttt{true}} = \frac{0.5-\alpha}{0.5}$ as shown in Figure 5.2. Thus, in the training process, we make $\frac{\texttt{false}}{\texttt{true}} = \frac{\beta|\mathbf{C}_{IR}^{(i)}|}{|\mathbf{C}_R^{(i)}|} = \frac{0.5-\alpha}{0.5}$. Then,

$$\beta = \lambda \frac{(1-2\alpha)|\mathbf{C}_R^{(i)}|}{|\mathbf{C}_{IR}^{(i)}|} \tag{5.3.4}$$

where $|\cdot|$ denotes the number of conditions and λ is a hyper-parameter to control the magnitude.

5.3.2. Adapter

Adapter has been shown to yield parameter-efficient tuning for NLP [**33**]. This ability is desirable to build a multi-skill dialogue model [**57**]. Several architectures of adapter have been proposed in the previous studies, as shown in Figure 5.3: added upon each transformer layer [**33, 57**], mixed within each layer [**62**], and added outside the pre-trained model [**93**]. We denote them as Top-, In-, Side-Adapter respectively. In the first two architectures, the adapter directly modifies the outputs of a transformer layer. In contrast, Side-Adapter will

[1]We first tried to use the predicted relevant conditions instead of the ground-truth for training in a certain proportion, but it damaged the performance.

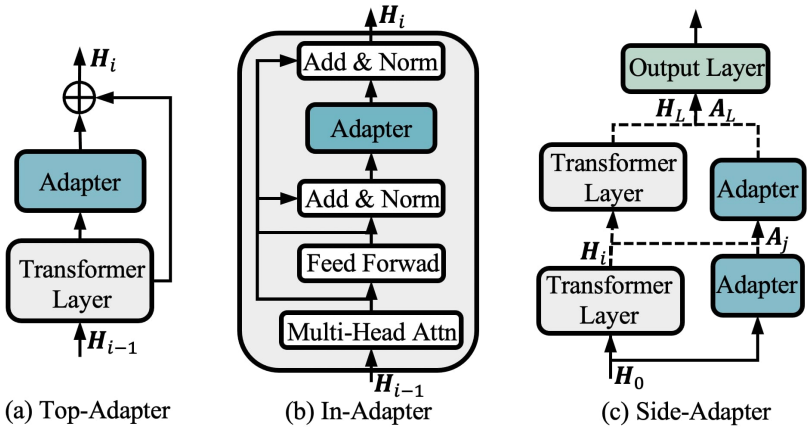

Fig. 5.3. Three types of adapter architectures explored in this work.

not modify the outputs of a pre-trained transformer, but utilizes an output layer to fuse two types of outputs at the end. Apart from the different positions, the components within an adapter are similar. An adapter layer is a down projection to a bottleneck dimension followed by an up projection to the initial dimension. Below, we only give the formulation of Top-Adapter.

Given the output of i-th Transformer layer $\mathbf{H}'_i \in \mathbb{R}^{n \times d}$, where n is the input length and d is the hidden dimension, $\mathbf{H}_i$ is:

$$\mathbf{H}_i = \texttt{ReLU}(\texttt{LN}(\mathbf{H}'_i)\mathbf{W}_i^D)\mathbf{W}_i^U + \mathbf{H}'_i, \tag{5.3.5}$$

where $\mathbf{W}_i^D \in \mathbb{R}^{d \times h}$, $\mathbf{W}_i^U \in \mathbb{R}^{h \times d}$, and $\texttt{LN}(\cdot)$ denotes layer normalization [**2**]. The bottleneck dimension h is tunable and it allows to adjust the capacity of the adapter.

5.3.3. Domain Adversarial Training

In addition to multi-task learning, we explore whether domain adversarial training [**23**] can remove domain-specific features and further help to capture common structure underlying all the datasets. This method has been shown to benefit both convolutional nets and recurrent nets [**48, 29**]. However, the work of [**49**] on clinical negation detection and that of [**98**] on multi-source domain adaptation showed that the method does not work well on pre-trained transformer, e.g. BERT. Recently, [**19**] demonstrated that it is effective to first encourage the model to be domain-aware and then conduct the domain adversarial training to derive the domain-invariant representations. We evaluate this idea in our experiments. As shown in Figure 5.1, we require the output of Feature Pooler (a linear layer with `Tanh` activation) to maximally confuse a domain classifier. This is accomplished through a min-max objective between the domain classifier θ_D and our model θ_G:

$$\mathcal{L}_{\text{D}}^{(i)} = \max_{\theta_G} \min_{\theta_D} - \sum_{k=1}^{|\mathbf{C}^{(i)}|} \log P_k(\mathbf{d}^{(i)} | \mathbf{x}^{(i)}, \mathbf{C}^{(i)}), \tag{5.3.6}$$

	PersonaChat	LIGHT
Train Set	33145	27325
Valid Set	4098	1782
Test Set	3738	3463

Table 5.1. Key characteristics of PersonaSkillTalk.

where $\mathbf{d}^{(i)}$ is the ground-truth domain of this sample. The effect of this objective is to improve the ability of the classifier to determine the domain of a sample, while encouraging the model to generate maximally confusing representations. In practice, this is implemented by training the model using standard loss, but reversing the gradients of the loss with respect to θ_G as indicated in Figure 5.1. We adopt the method of [**19**], and thus before domain adversarial training we first remove the Gradient Reversal layer and train the model to be domain-aware.

5.4. Experiments

5.4.1. Data Collection

We build our experiment dataset, named **PersonaSkillTalk** based on two public datasets: PersonaChat [**108**] and LIGHT [**90**], where dialogues are both conditioned on persona profiles. Meanwhile, the two datasets are in clearly different domains: the former is related to daily life while the latter involves interactions between characters (or animals) in a text adventure game. Thus, if a model mainly learns domain-specific features on a dataset, transferring it to another will not yield good performance.

We automatically annotate whether a persona profile has been employed in the ground-truth response based on simple matching as follows [2]: if some key word in a persona profile match the key words in the response, then we consider that the persona profile is used. An automatic annotation is possible in this case because the sentences to construct persona profiles are simple, such as "I am a student." or "I love dogs.", and the vocabulary of persona profiles is much smaller than the vocabulary of dialogues. The automatic annotation lead to a very high accuracy: A manual evaluation of 200 random samples reveals that 92.3% of the automatic annotations conform to human judgement. We can thus use the annotations in model training. To fully evaluate the performance on modeling dialogue skill, we only keep samples where the response is constructed based on at least a persona profile, which produces 40,981 (51%) and 32,570 (28%) samples for PersonaChat (denoted **PC**) and LIGHT (denoted **LI**) dataset respectively as in Table 5.1.

[2]We use lemmatization, stop word removal, and rules considering the results of part-of-speech tagging to identify key words. We will release the code to construct the dataset.

5.4.2. Baselines

Our main experiments are: 1) validating the effectiveness of the auxiliary loss (denoted as $\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising) comparing to the response reconstruction objective employed by the previous works (denoted as $\mathcal{L}_Y$ only) [**86, 77, 87, 57**]; 2) exploring whether multi-task learning and domain adversarial training help to learn a domain-independent skill; 3) investigating the functionality of adapter.

Specifically, with two different training objectives, we compare: fine-tuning on the target dataset (denoted as **FT**) [**86, 77**], first training on another dataset then fine-tuning on the target dataset (**X-FT+FT**) [**86**], multi-task learning on the two datasets (**MultiT**) [**86, 87**], only fine-tuning the adapter (**Adap**) [**57**], multi-task learning using the adapter (**MultiT-Adap**), and applying domain adversarial training (**Dinv-Adap**).

5.4.3. Implementation Details

Since PersonaSkillTalk is a small dataset, the base dialogue model in our experiments (all models using this pre-trained dialogue model) is a small-scale Transformer-AR (110M) that uses bi-directional attention on the source side and left-to-right attention on the target side [**106**]. The model is initialized with BERT (base, uncased) [**15**] and has been further pre-trained on 8M dialogue dataset consisting of Reddit [**20**] [3] and Twitter [4] data that has been carefully preprocessed.

For the proposed auxiliary loss, we do hyper-parameter search based on the performance on the validation set for both α in $\{0.2, 0.3, 0.4, 0.5\}$ and λ, the magnitude of β, in $\{0.2, 0.3, 0.5, 1.0\}$. We report experimental results with $\alpha = 0.3$ and $\lambda = 0.2$. For the ablation study of training without denoising, α is set to 0.5. We use a P100 GPU for training. The batch size is 20, and the maximum input length is set to 256. We apply early stopping according to the performance on the validation set. For decoding, the beam size is 4. We prevent duplicated uni-grams, and set minimum response length to encourage diverse generation as in [**77**]: The minimum response length is set to make the average length of generated responses match with the average target length of the dataset.

When comparing different adapter architectures, we vary the bottleneck dimension of Top-Adapter and In-Adapter to change the model size. For Side-Adapter, we vary both the bottleneck dimension and the number of Transformer layers inside each Adapter. We implement Top-Adapter and In-Adapter, and take the open-source code of Side-Adapter [**93**][5].

[3] `https://github.com/nouhadziri/THRED`
[4] `https://github.com/Marsan-Ma-zz/chat_corpus`
[5] `https://github.com/microsoft/k-adapter`

		PPL		R-Acc		C-Hit	
		LI	PC	LI	PC	LI	PC
$\mathcal{L}_Y$ only	FT	**23.4**	**13.4**	-	-	63.0	**46.6**
	X-FT + FT	**23.4**	**13.4**	-	-	**63.3**	45.6
	MultiT	24.1	13.9	-	-	58.3	43.0
	Adapter	25.2	14.3	-	-	60.9	45.1
	MultiT-Adap	25.2	14.4	-	-	60.0	43.8
$\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising (ours)	FT	24.5	15.2	65.2	63.2	66.5	56.3
	X-FT + FT	**24.2**	**14.8**	**65.8**	63.3	**68.4**	54.2
	MultiT	24.6	**14.8**	65.4	63.2	66.8	56.2
	Adapter	24.7	15.8	65.1	63.2	64.1	56.4
	MultiT-Adap	26.1	15.8	65.2	**63.9**	**68.4**	**57.0**
	Dinv-Adap	25.8	15.3	64.9	63.2	67.2	56.3

Table 5.2. Performance on PersonaSkillTalk. We report the results on LIGHT (LI) and PersonaChat (PC) respectively. The upper half is training with $\mathcal{L}_Y$ only, and the lower half is using our training approach.

		Cohe.	Appr.
$\mathcal{L}_Y$ only	FT	0.57	1.14
	X-FT + FT	0.51	0.94
	MultiT	0.51	0.90
$\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising (ours)	FT	0.61	1.27
	X-FT + FT	0.63	1.22
	MultiT	0.65	1.10
	MultiT-Adap	**0.67**	**1.33**

Table 5.3. Human evaluation on response coherence (Cohe.) and condition appropriateness (Appr.).

5.4.4. Evaluation

Automatic Metrics. We compare model performance using: 1) perplexity (**PPL**) that is a reformulation of the standard training objective $\mathcal{L}_Y$ reflecting how well the model fit the dataset (i.e. measuring how likely the model generates the ground-truth responses); 2) the accuracy (%) of relevant condition recognition (**R-ACC**); 3) the percentage (%) of at least 1 ground-truth relevant condition detected in the generated responses (**C-HIT**). We use the automatic script used in the data collection process to evaluate C-HIT.

Human Evaluation. Furthermore, we ask human evaluators to rate whether a response is coherent to the dialogue history in $\{0, 1\}$, denoted as **Coherence**. 1 represents a fluent and coherent response. We also evaluate whether a response utilizes correct condition(s) that is also relevant to the dialogue history in {0,1,2}, denoted as **Condition Appropriateness**. 1 means the response uses correct condition(s); 2 means the utilized condition(s) is correct

and relevant. We only perform human evaluation on parts of the models that have shown good performance given automatic metrics. The test set consists of 100 random samples in PersonaChat and 100 in LIGHT. The inter-rater annotation agreement in Cohen's kappa [**14**] is 0.44 and 0.71 for coherence and condition appropriateness, which indicates moderate and substantial agreement.

Dialogue History		i am male , divorce , 6 grandchildren , ex named doug .
Persona Profiles		i graduated from college last year . (0)
		i work in an office as a secretary . (0)
		my name is lucy . (1)
		i like to eat candy . (0)
		i was on the soccer team in college . (0)
Gold Response		nice to meet you doug . i am lucy
$\mathcal{L}_Y$	FT	that is great . i am a secretary in college (1, 0)
	X-FT + FT	hi i am lucy , my name is . how are ? that sucks (0, 0)
	MultiT	that is cool i am a secretary my name lucy (1, 2)
$\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising	FT	that is nice i am a secretary in office now (1, 2)
	X-FT + FT	hi i am lucy . that is cool my name was in college (0, 0)
	MultiT	that s cool i m a college student my name is lucy what do you like (1, 0)
	MultiT + Adap	that s cool i m a secretary my name lucy (1, 2)

Table 5.4. Generated responses on **PersonaChat**. The automatic annotations, i.e. (0/1), are given at the end of each persona profile. Human evaluation scores are given at the end of each generated reply, representing (response coherence, condition appropriateness).

Dialogue History		me either . not since i quit working there and began writing short stories . [SEP] what are some of your stories like ?
Persona Profiles		i like dark superhero movies . (1)
		i like to drink scotch to relax . (0)
		i write short stories for a living . (0)
		i used to work at a carnival . (0)
Gold Response		i get alot of my inspiration from superhero movies . little mix of all my faves .
$\mathcal{L}_Y$	FT	i love superhero . my stories are dark like a ghost (0, 2)
	X-FT + FT	i like dark superhero . what do you write ? (0, 2)
	MultiT	i like dark superhero movies . they are my favorite (0, 2)
$\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising	FT	mostly superhero . i write a lot about drinking scotch (1, 2)
	X-FT + FT	i am a superhero writer and love stories like the dark ones (1, 2)
	MultiT	i am a writer . mostly horror stories and thrillers (1, 2)
	MultiT + Adap	mostly superhero . i love them especially the dark ones (1, 2)

Table 5.5. Generated responses on **PersonaChat**. The automatic annotations, i.e. (0/1), are given at the end of each persona profile. Human evaluation scores are given at the end of each generated reply, representing (response coherence, condition appropriateness).

Dialogue History		what brings you to the king ' s throne room this day ?
Persona Profiles		the neighboring kingdom has been destroyed . (1) what was once a castle is now smoldering ruin . (1) i have to warn him , before they are next . (0) i have arrived after days of travel with grave news for the king . (1)
Gold Response		your majesty , i have terrible news of dire consequence . the neighboring kingdom has been over - run . the castle is burned to the grounds and even know it smoulders , nothing more than a pile of rubble . this scroll gives a full account of what transpired .
$\mathcal{L}_Y$	FT	i am looking for new to the kingdom . my father is king and he has been here years (1, 0)
	X-FT + FT	i am here to warn the king he is looking for treasure (1, 0)
	MultiT	i am here to warn the king . he is looking for a new kingdom that has been destroyed (1, 0)
$\mathcal{L}_Y + \mathcal{L}_R$	FT	i am here to news of the loss our kingdom s king (0, 0)
	X-FT + FT	i am looking for news the king has been . he is having a talk with (0, 0)
w/ denoising	MultiT	i am here to the king s quarters this morning my kingdom has been destroyed (1, 2)
	MultiT + Adap	i am looking for news from the king s quarters how are things in kingdom (1, 0)

Table 5.6. Generated responses on **LIGHT**. The automatic annotations, i.e. (0/1), are given at the end of each persona profile. Human evaluation scores are given at the end of each generated reply, representing (response coherence, condition appropriateness).

5.4.5. Results and Analysis

$\mathcal{L}_Y$ only and Untransferable Skill Our experimental results in Table 5.2 show that X-FT + FT performs very similarly as FT, indicating that leveraging another dataset (e.g. using PersonaChat for LIGHT) does not help. Multi-task learning should have helped the model to learn the common structure underlying the two datasets, here the persona skill; but MultiT performs worse than FT in all automatic metrics especially in C-Hit, which means it weakens the ability to utilize relevant conditions in generation. Human evaluation results in Table 5.3 ($\mathcal{L}_Y$ only) also show that leveraging another dataset weakens model performance. Similar observation has already been reported in a previous work that applies $\mathcal{L}_Y$ only [**86**]. The reason behind these phenomena can only be that by reconstructing the ground-truth responses ($\mathcal{L}_Y$ only), the model also learns domain-specific features in additional to the underlying dialogue skill. Thus, leveraging another dataset of a clearly different domain does not only transfer the knowledge of the dialogue skill to the target dataset, but also brings noises in the training on the target dataset, and thus damages the model performance.

$\mathcal{L}_R$ and Denoising Training As summarized in the lower half of Table 5.2, with the proposed training objective, X-FT + FT performs better than FT in most automatic metrics,

Dialogue History		ah , this choir room is amazing .
Persona Profiles		i am financed by the king , so i often work in the palace , making art for the royals . (0) working on my art makes me extremely happy . (0) i love to paint the scenery of my village , the beautiful views of nature around me . (1)
Gold Response		it sure is beautiful . it ' s one of my favorite areas to paint .
$\mathcal{L}_Y$	FT	yes it is a beautiful room i love the view (1, 0)
	X-FT + FT	yes it is my work i love painting the scenery (1, 2)
	MultiT	yes it is i love my paintings what do you (0, 2)
$\mathcal{L}_Y + \mathcal{L}_R$	FT	yes it is my pleasure i love the view you are very (0, 0)
	X-FT + FT	it is indeed . i am very proud of my work , the king has me a great artist (1, 2)
w/ denoising	MultiT	it s a beautiful room indeed i am very proud of my work (1, 2)
	MultiT + Adap	it s a beautiful place i am painting the scenery you d like my majesty (1, 2)

Table 5.7. Generated responses on **LIGHT**. The automatic annotations, i.e. (0/1), are given at the end of each persona profile. Human evaluation scores are given at the end of each generated reply, representing (response coherence, condition appropriateness).

and MultiT performs very similarly to FT. Although the improvement is small, so as the corresponding human evaluation results in Table 5.3, comparing with $\mathcal{L}_Y$ only, our training approach indeed prevents leveraging another dataset of a different domain from damaging the model performance on the target task. Particularly, MultiT-Adap performed much worse than FT, but with our training approach it performs better than FT in both automatic and manual evaluations. It is thus viewed as a promising avenue to learn a transferable dialogue skill. We will further discuss it subsequently.

When comparing the model trained with our training approach to a model with $\mathcal{L}_Y$ only, the results show that our approach largely enhances the ability of utilizing relevant conditions in generation (C-Hit and Appr.). It thus confirms that an auxiliary loss additional to NLLLoss enables the model to focus on learning the dialogue skill instead of general expressions related to a domain. However, we indeed observe a small increase in perplexity. It is expected since perplexity is related to $\mathcal{L}_Y$, and thus training on $\mathcal{L}_Y$ only will yield lower perplexity.

Denoising is the core component of the proposed training approach. It simulates the noisy situations when the predicted relevant conditions are inaccurate (low R-ACC). To validate its effectiveness, Table 5.8 reports the ablation study of training without denoising. The last row summarizes its average decrease in performance comparing with $\mathcal{L}_Y + \mathcal{L}_R$ w/ denoising. The results show that denoising training enables the model to better fit the datasets (lower perplexity) and enhances the ability of utilizing relevant conditions in generation (higher C-Hit).

	PPL		C-Hit	
	LI	PC	LI	PC
FT	**24.5**	**16.8**	64.5	54.2
X-FT + FT	24.7	17.0	**65.4**	53.6
MultiT	25.1	17.1	**65.4**	**55.6**
Adapter	25.2	17.6	61.3	55.1
MultiT-Adap	28.1	18.9	62.2	53.0
avg.	+0.7	+2.2	-3.1	-2.3

Table 5.8. Ablation study of w/o denoising. The last row summarizes the average decrease in performance.

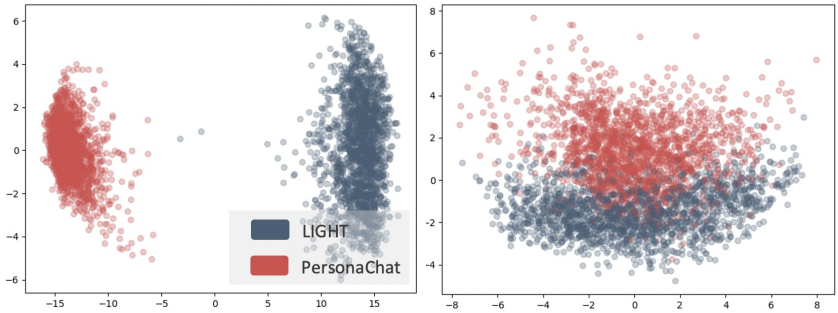

Fig. 5.4. PCA visualization of the outputs of Feature Pooler on the test set without and with domain adversarial training.

Multi-task Learning and Domain Adversarial Training As discussed before, when training the model using only response reconstruction loss, multi-task learning on two datasets in clearly different domains substantially damages the performance on each task. In contrast, with our training approach, multi-task learning on average improves the performance. We expect that with more persona-skill datasets, the multi-task learning can better capture the skill underlying these tasks. In PersonaSkillTalk, there are only two datasets that are in clearly different domains, which poses much difficulty to learn the common structure. Nevertheless, we find that decreasing model capacity helps to learn the common part. We will further discuss it later.

In addition to multi-task learning, we explore whether direct domain adversarial training helps the model to be domain-independent. As shown in Figure 5.4, with domain adversarial training, the distance between the data points from different domains becomes much smaller. However, the performance in generation is not improved (see Table 5.2). We also observe that the adversarial training process can be unstable as reported in the previous work [**98**]. These show that multi-task learning is a more effective and robust way to learn the underlying dialogue skills.

Multi-task Learning with Adapter Adapter training a small set of parameters has been shown to have comparable performance to fine-tuning all parameters [**57**]. In this

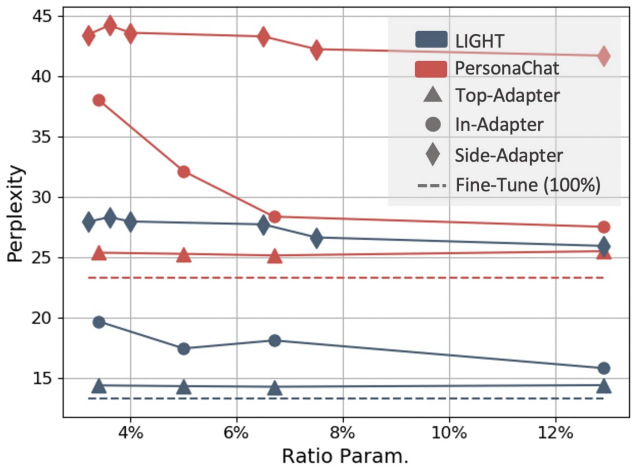

Fig. 5.5. Performance comparison among three types of adapters on the two datasets. The x-axis is the size of adapter comparing to the size of the base model.

work, we compare the three adapter architectures as well as fine-tuning all parameters using $\mathcal{L}_Y$ only, denoted as Fine-Tune (100%), and report the results in Figure 5.5. As expected, we can observe that fine-tuning the entire model performs the best. Nevertheless, Top-Adapter can consistently get closer to the performance of fine-tuning with only $\sim 6\%$ task-specific parameters. Increasing the adapter size does not obviously affect Top-Adapter but improves the performance of In-Adapter. Only a large In-Adapter ($\sim 13\%$ of parameters) can yield comparative performance to Top-Adapter. We also see that Side-Adapter cannot efficiently adapt the pre-trained dialogue model to a specific dialogue task and performs the worst. Therefore, Top-Adapter is most suitable to build a single-skill dialogue model. The results reported in Table 5.2 have been produced using Top-Adapter (fine-tuning only 6.7% parameters).

What is discovered in our experiments is that with our training approach MultiT-Adap outperforms both FT and MultiT in both automatic (except in terms of perplexity) and manual evaluation, indicating that MultiT-Adap best learns the common dialogue skill underlying the two datasets. We assume the behind reason is that the much smaller capacity of adapter (comparing with fine-tuning all parameters) helps to distill a domain-independent skill since it is more undisturbed by diverse domain-specific features. The experiment results of $\mathcal{L}_Y$ only in Table 5.2 also indicate that MultiT-Adap learns less domain-specific features than MultiT (higher PPL) but better ability of utilizing relevant conditions in generation (higher C-Hit).

5.5. Conclusion

In this chapter, we investigated the problem of dialogue skill modeling. Different from previous studies, we focus on the transferability of a skill model. Through experiments on

PersonaSkillTalk, we confirmed that using NLLLoss only makes the model capture everything that is specific to the dataset, including the skill and domain-specific features. This explains why multi-task learning performed worse than fine-tuning on each dataset separately, and weakened the ability to utilize relevant conditions in generation. Instead, our experiments revealed a promising avenue toward learning transferable dialogue skills, including:

- An auxiliary task specifically designed to extract a skill is necessary. We proposed a training approach that first requires the model to explicitly predict whether a condition is relevant to dialogue history so that the model should use it in generation. Next, we trained the model by conditioned response generation task based on these relevant conditions. Experimental results showed that with our training approach, multi-task learning improves the model performance. Our approach also largely enhances the general ability of utilizing relevant conditions in generation.
- Multi-task learning is effective to capture the common dialogue skill underlying the datasets with our training objective. However, domain adversarial training [**23**] that is used for removing domain-specific features does not further help to learn a domain-independent skill.
- Adapter helps to distill a domain-independent skill. An adapter is a small set of trainable parameters that steer a fixed base model to a down-stream task. Previous work has shown that it yields parameter-efficient tuning for NLP [**33**]. Our experiments revealed its advantage on learning dialogue skills: its much smaller capacity (comparing with fine-tuning all parameters) prevents it from learning diverse domain-specific features.

Chapter 6

Conclusion and Future Work

6.1. Overview

In summary, this dissertation presented a series of methods to leverage pre-trained language models for both task-oriented dialogue systems and general-domain chatbots. Particularly, we focused on the latter and provided solutions starting from general open-domain dialogue generation to conditioned dialogue generation.

Pre-trained language models have shown to be effective for improving many natural language processing tasks. For task-oriented dialogue systems, the existing state-of-the-art generative framework in DST replaces RNN encoder with BERT. However, this framework still utilizes an RNN decoder stacked upon BERT encoder. In Chapter 2, we proposed a framework consisting of a single BERT that works as both the encoder and the decoder, which has a flat encoder-decoder architecture allowing for more effective parameter updating. Experiments on MultiWOZ datasets showed that our model substantially outperforms the existing framework, and it also achieves very competitive performance to the best ontology-based approaches. Besides, it can converge to its best performance much faster and in a more stable manner than the existing framework.

For generative chatbot systems, GPT, a left-to-right language model, has shown to generate fluent and diverse text, and thus previous works fine-tuned GPT for open-domain dialogue generation. Nevertheless, some works showed that fine-tuning BERT can also achieve state-of-the-art performance. Thus, we investigated this problem and examined how to best exploit a pre-trained LM for dialogue generation in Chapter 3. Specifically, we compared 4 frameworks that utilize pre-trained LM for open-domain dialogue generation on 3 public datasets each in large and small scale. The comparison revealed that Transformer-Dec and Transformer-AR are both good choices when large-scale data is available, e.g. further dialogue pre-training. When data is limited, e.g. fine-tuning on small dialogue tasks, Transformer-Dec is the most appropriate. Through extensive experiments, we also observed the impact of pretrain-finetune discrepancy and finetune-generation discrepancy, and we examined the

discrepancies of each framework. Further, we proposed two novel methods to reduce discrepancies, yielding improved performance. This study is the first investigation on the widely used 4 frameworks based on pre-trained LM in terms of architectural appropriateness and discrepancies.

Beyond the pre-trained Transformer based frameworks for task-oriented dialogue and open-domain dialogue, we go further in conditioned dialogue generation. In Chapter 4, we proposed a simple and efficient multi-task learning approach for loosely-conditioned response generation to alleviate the data scarcity issue of labeled dialogues. We show that labeled text (non-dialogue) data that are much easier to collect can supplement labeled dialogue data. These data can be, for example, texts written by the same person (for a persona condition), within the same topic domain (for a topic condition), etc. We confirmed that these data can contribute to create better representations of conditions and better utilization of conditions in natural language generation. Our approach outperforms the state-of-the-art models by leveraging labeled texts and obtains larger improvement in performance compared to the existing methods to leverage text data.

We also investigate how to equip a chatbot with dialogue skills in Chapter 5. We showed that Adapter has enough capacity to model a dialogue skill while needing only 6% more parameters than a pre-trained dialogue model. Thus, it is possible to build a multi-skill model by using a fixed base model and multiple small Adapters. However, we found that previous works in dialogue skill learning also learn domain-specific expressions. In this case, combing two dialogue "skills" does not improve the performance due to the non-transferablility of these domain-specific features. Thus, we proposed methods for learning a dialogue skill by: 1) auxiliary loss specifically designed for the skill; 2) multi-task learning to learn the common part, i.e. skill, of several tasks; 3) Adapter to decrease model capacity avoiding to learn diverse styles. Experimental results show that with our approach a model more likely generates appropriately conditioned responses.

At last, we summarize our works in the following table:

Task-oriented Dialogue Systems	A flat encoder-decoder framework consisting of a single BERT
Generative Chatbots	
Open-domain dialogue	1) Transformer-ED/Dec/MLM/AR; 2) Pretrain-finetune and finetune-generation discrepancy; 3) Two methods to decrease model discrepancies
Conditioned dialogue	1) A multi-task learning approach for loosely conditioned generation; 2) Multi-skill chatbots by a fixed base pre-trained model & Adapters; 3) Methods of learning transferable dialogue skills

Table 6.1. A summary of the proposed methods that better exploit pre-trained language model for dialogue systems.

6.2. Future Research Directions

This thesis suggests many promising future research directions.

- **Language Modeling** Before the rise of pre-trained Transformer-based language models, works in dialogue generation were all based on RNN Seq2Seq framework. Many methods had been proposed and achieved better performance than vanilla RNN, e.g. hierarchical RNN [**81**] for multi-turn response generation, CVAE [**110**] to generate diverse responses, CCM [**114**] leveraging knowledge graph for response generation, generation under different conditions, and so on. However, none of these works outperforms vanilla pre-training based models that do not exploit extra knowledge. Furthermore, many previous ideas that work well in RNN based systems might not bring similar improvement to a Transformer based system. For example, multi-turn response generation in a Transformer based system does not apply a hierarchical model structure. Instead, it simply concatenates multi-turn dialogue history as the input of Transformer. Therefore, we believe that future improvement in dialogue generation will be still from the improvement in language model. The question will be how to continue scaling a language model and how to efficiently train such a giant model on extremely large-scale text corpus.
- **Conditioned Generation** More and more attention is paid to controlling language generation. Researchers are interested in controlling the style, topic, knowledge, and so on. Some of these approaches for dialogue generation specifically have been described in Chapter 4, e.g. using a parametric vector or Adapter. However, generation based on scheme is a more challenging task. To generate a long text, e.g. a story, researchers usually want a generative model to expand the story based on a scheme. The scheme controls the logic and the main story, and the model enables the diversity of generated text.
- **Content to be Generated** Much of the current research work tries to generate a response from an internal representation built from the dialogue history. The assumption is that a powerful language model would be able to determine what to generate. This is difficult in practice. In many real situations, we need to control tightly the content of the generation. There is a need to combine the flexible neural framework used in the current research on dialogue generation with the strategy used in traditional dialogue generation: determine first what to generate, then how to generate. More investigations are required to determine the content to be generated.
- **Multi-Skill System** For dialogue systems, researchers have realized that a vanilla Seq2Seq model for open-domain dialogue generation overly simplifies the task: it leaves a model to generate a response given dialogue history without extra knowledge. Instead, recent studies have focused on modeling dialogue skills and how to

switch among skills given a dialogue history using elaborate rules or more explainable models. There are always debates on whether to use a unified solution or a specifically-designed one for NLP tasks. We believe that compromise will be a better choice. For example, BERT is a unified solution and different LM heads, e.g. linear layers, stacked on it for different NLU tasks have achieved superior performance. However, Google T5 applying the idea of "text-to-text" to incorporate most NLP tasks may be overly unified. Thus, we believe that the future framework will consist of a powerful base model and multiple small adapters for each target task.

References

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu *et al.* : Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Jimmy Lei Ba, Jamie Ryan Kiros et Geoffrey E Hinton : Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou *et al.* : Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*, 2020.

[4] Siqi Bao, Huang He, Fan Wang et Hua Wu : Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv:1910.07931*, 2019.

[5] Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu et Xinchao Xu : Plato-2: Towards building an open-domain chatbot via curriculum learning. *arXiv preprint arXiv:2006.16779*, 2020.

[6] Ankur Bapna et Orhan Firat : Simple, scalable adaptation for neural machine translation. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548, 2019.

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent et Christian Janvin : A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

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

[9] Rich Caruana : Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[10] Nathanael Chambers et James Allen : Stochastic language generation in a dialogue system: Toward a domain independent generator. Rapport technique, FLORIDA INSTITUTE FOR HUMAN AND MACHINE COGNITION INC PENSACOLA FL, 2004.

[11] Chaotao Chen, Jinhua Peng, Fan Wang, Jun Xu et Hua Wu : Generating multiple diverse responses with multi-mapping and posterior mapping selection. *arXiv preprint arXiv:1906.01781*, 2019.

[12] Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer *et al.* : Gunrock: Building a human-like social bot by leveraging large scale real user data. *Alexa Prize Proceedings*, 2018.

[13] Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan et Kai Yu : Schema-guided multi-domain dialogue state tracking with graph attention neural networks. *In AAAI*, pages 7521–7528, 2020.

[14] Jacob Cohen : A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee et Kristina Toutanova : Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe *et al.* : The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*, 2019.

[17] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli et Jason Weston : Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

[18] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou et Hsiao-Wuen Hon : Unified language model pre-training for natural language understanding and generation. *In Advances in Neural Information Processing Systems*, pages 13042–13054, 2019.

[19] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi et Jianxin Liao : Adversarial and domain-aware bert for cross-domain sentiment analysis. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, 2020.

[20] Nouha Dziri, Ehsan Kamalloo, Kory Mathewson et Osmar R Zaiane : Augmenting neural response generation with context-aware topical attention. *In Proceedings of the First Workshop on NLP for Conversational AI*, pages 18–31, 2019.

[21] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao et Dilek Hakkani-Tur : Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

[22] Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao et Bill Dolan : Grounded response generation task at dstc7. *In AAAI Dialog System Technology Challenges Workshop*, 2019.

[23] Yaroslav Ganin et Victor Lempitsky : Unsupervised Domain Adaptation by Backpropagation. *In International Conference on Machine Learning*, pages 1180–1189, 2015.

[24] Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li et Shuming Shi : Generating multiple diverse responses for short-text conversation. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6383–6390, 2019.

[25] Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung et Dilek Hakkani-Tur : Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*, 2019.

[26] Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao et Bill Dolan : Jointly optimizing diversity and relevance in neural response generation. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1229–1238, 2019.

[27] Rahul Goel, Shachi Paul et Dilek Hakkani-Tür : Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *arXiv preprint arXiv:1907.00883*, 2019.

[28] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha et Sunghun Kim : Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*, 2018.

[29] Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng et Xuan-Jing Huang : Part-of-Speech Tagging for Twitter with Adversarial Neural Networks. *In EMNLP 2017*, pages 2411–2420, 2017.

[30] Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi et Milica Gašić : Trippy: A triple copy strategy for value independent neural dialog state tracking. *arXiv preprint arXiv:2005.02877*, 2020.

[31] Matthew Henderson, Blaise Thomson et Steve Young : Word-based dialog state tracking with recurrent neural networks. *In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, 2014.

[32] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz et Richard Socher : A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*, 2020.

[33] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan et Sylvain Gelly : Parameter-efficient transfer learning for NLP. *In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2790–2799, 2019.

[34] Jiaying Hu, Yan Yang, Chencai Chen, Zhou Yu *et al.* : Sas: Dialogue state tracking via slot attention and slot information sharing. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6366–6375, 2020.

[35] Charles Lee Isbell, Michael Kearns, Dave Kormann, Satinder Singh et Peter Stone : Cobot in lambdamoo: A social statistics agent. *In AAAI/IAAI*, pages 36–41, 2000.

[36] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark et Hannaneh Hajishirzi : Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020.

[37] Sungdong Kim, Sohee Yang, Gyuwan Kim et Sang-Woo Lee : Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*, 2019.

[38] Brian Langner, Stephan Vogel et Alan W Black : Evaluating a dialog language generation system: comparing the mountain system to other nlg approaches. *In Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[39] Hung Le, Richard Socher et Steven CH Hoi : Non-autoregressive dialog state tracking. *arXiv preprint arXiv:2002.08024*, 2020.

[40] Hwaran Lee, Jinsik Lee et Tae-Yoon Kim : Sumbt: Slot-utterance matching for universal and scalable belief tracking. *arXiv preprint arXiv:1907.07421*, 2019.

[41] Jason Lee, Kyunghyun Cho et Thomas Hofmann : Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378, 2017.

[42] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He et 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.

[43] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao et Bill Dolan : A diversity-promoting objective function for neural conversation models. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, 2016.

[44] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao et Bill Dolan : A persona-based neural conversation model. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, 2016.

[45] Jiwei Li et Dan Jurafsky : Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*, 2016.

[46] Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho et Jason Weston : Don't say that! making inconsistent dialogue unlikely with unlikelihood training. *arXiv preprint arXiv:1911.03860*, 2019.

[47] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao et Shuzi Niu : Dailydialog: A manually labelled multi-turn dialogue dataset. *In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, 2017.

[48] Yitong Li, Timothy Baldwin et Trevor Cohn : What's in a domain? learning domain-robust text representations using adversarial training. *In Proceedings of NAACL-HLT*, pages 474–479, 2018.

[49] Chen Lin, Steven Bethard, Dmitriy Dligach, Farig Sadeque, Guergana Savova et Timothy A Miller : Does BERT Need Domain Adaptation for Clinical Negation Detection? *Journal of the American Medical Informatics Association*, 27(4):584–591, 2020.

[50] Zhaojiang Lin, Andrea Madotto et Pascale Fung : Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020.

[51] Zhaojiang Lin, Peng Xu, Genta Indra Winata, Zihan Liu et Pascale Fung : Caire: An end-to-end empathetic chatbot. *arXiv preprint arXiv:1907.12108*, 2019.

[52] Bing Liu et Ian Lane : End-to-end learning of task-oriented dialogs. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73, 2018.

[53] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser et Noam Shazeer : Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

[54] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer et Veselin Stoyanov : Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[55] Ryan Thomas Lowe, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu et Joelle Pineau : Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65, 2017.

[56] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao et Michel Galley : Multi-task learning for speaker-role adaptation in neural conversation models. *In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 605–614, 2017.

[57] Andrea Madotto, Zhaojiang Lin, Yejin Bang et Pascale Fung : The adapter-bot: All-in-one controllable conversational model. *arXiv preprint arXiv:2008.12579*, 2020.

[58] Michael McCloskey et Neal J Cohen : Catastrophic interference in connectionist networks: The sequential learning problem. *In Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[59] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson et Steve Young : Neural belief tracker: Data-driven dialogue state tracking. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, 2017.

[60] Tong Niu et Mohit Bansal : Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389, 2018.

[61] Kishore Papineni, Salim Roukos, Todd Ward et Wei-Jing Zhu : Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[62] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho et Iryna Gurevych : Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

[63] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych et Sebastian Ruder : Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*, 2020.

[64] Jason Phang, Thibault Févry et Samuel R Bowman : Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*, 2018.

[65] Alec Radford, Karthik Narasimhan, Tim Salimans et Ilya Sutskever : Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

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

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

[68] Osman Ramadan, Paweł Budzianowski et Milica Gasic : Large-scale multi-domain belief tracking with knowledge sharing. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, 2018.

[69] Owen Rambow, Srinivas Bangalore et Marilyn Walker : Natural language generation in dialog systems. Rapport technique, AT AND T LABS-RESEARCH FLORHAM PARK NJ, 2001.

[70] Hannah Rashkin, Eric Michael Smith, Margaret Li et Y-Lan Boureau : Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*, 2018.

[71] Hannah Rashkin, Eric Michael Smith, Margaret Li et Y-Lan Boureau : Towards empathetic open-domain conversation models: A new benchmark and dataset. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, 2019.

[72] Adwait Ratnaparkhi : Trainable methods for surface natural language generation. *In 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000.

[73] Sylvestre-Alvise Rebuffi, Hakan Bilen et Andrea Vedaldi : Learning multiple visual domains with residual adapters. *In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 506–516, 2017.

[74] Liliang Ren, Jianmo Ni et Julian McAuley : Scalable and accurate dialogue state tracking via hierarchical sequence generation. *arXiv preprint arXiv:1909.00754*, 2019.

[75] Liliang Ren, Kaige Xie, Lu Chen et Kai Yu : Towards universal dialogue state tracking. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786, 2018.

[76] Alan Ritter, Colin Cherry et William B Dolan : Data-driven response generation in social media. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, 2011.

[77] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith *et al.* : Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[78] Shoetsu Sato, Naoki Yoshinaga, Masashi Toyoda et Masaru Kitsuregawa : Modeling situations in neural chat bots. *In Proceedings of ACL 2017, Student Research Workshop*, pages 120–127, 2017.

[79] Abigail See, Stephen Roller, Douwe Kiela et Jason Weston : What makes a good conversation? how controllable attributes affect human judgments. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723, Minneapolis, Minnesota, juin 2019. Association for Computational Linguistics.

[80] Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke *et al.* : A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*, 2017.

[81] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville et Joelle Pineau : Building end-to-end dialogue systems using generative hierarchical neural network models. *In Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[82] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville et Yoshua Bengio : A hierarchical latent variable encoder-decoder model for generating dialogues. *In Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[83] Yong Shan, Zekang Li, Jinchao Zhang, Fandong Meng, Yang Feng, Cheng Niu et Jie Zhou : A contextual hierarchical attention network with adaptive objective for dialogue state tracking. *arXiv preprint arXiv:2006.01554*, 2020.

[84] Xiaoyu Shen, Hui Su, Shuzi Niu et Vera Demberg : Improving variational encoder-decoders in dialogue generation. *In Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[85] Kurt Shuster, Samuel Humeau, Antoine Bordes et Jason Weston : Engaging image chat: Modeling personality in grounded dialogue. *arXiv preprint arXiv:1811.00945*, 2018.

[86] Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau et Jason Weston : The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. *arXiv preprint arXiv:1911.03768*, 2019.

[87] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston et Y-Lan Boureau : Can you put it all together: Evaluating conversational agents' ability to blend skills. *arXiv preprint arXiv:2004.08449*, 2020.

[88] Ronnie W Smith et D Richard Hipp : *Spoken natural language dialog systems: A practical approach.* Oxford University Press on Demand, 1994.

[89] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao et Bill Dolan : A neural network approach to context-sensitive generation of conversational responses. *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, 2015.

[90] Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam et Jason Weston : Learning to speak and act in a fantasy text adventure game. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683, 2019.

[91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser et Illia Polosukhin : Attention is all you need. *In Advances in neural information processing systems*, pages 5998–6008, 2017.

[92] Ramakrishna Vedantam, C Lawrence Zitnick et Devi Parikh : Cider: Consensus-based image description evaluation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[93] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou *et al.* : K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[94] Joseph Weizenbaum : Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[95] Sean Welleck, Jason Weston, Arthur Szlam et Kyunghyun Cho : Dialogue natural language inference. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy, juillet 2019. Association for Computational Linguistics.

[96] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes et Steve Young : A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[97] Thomas Wolf, Victor Sanh, Julien Chaumond et Clement Delangue : Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.

[98] Dustin Wright et Isabelle Augenstein : Transformer based multi-source domain adaptation. *arXiv preprint arXiv:2009.07806*, 2020.

[99] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher et Pascale Fung : Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*, 2019.

[100] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou et Wei-Ying Ma : Topic aware neural response generation. *In Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[101] Puyang Xu et Qi Hu : An end-to-end approach for handling unknown slot values in dialogue state tracking. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457, 2018.

[102] Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope et Ray Kurzweil : Learning semantic textual similarity from conversations. *In Proceedings of The Third Workshop on Representation Learning for NLP*, pages 164–174, Melbourne, Australia, juillet 2018. Association for Computational Linguistics.

[103] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov et Quoc V Le : Xlnet: Generalized autoregressive pretraining for language understanding. *In Advances in neural information processing systems*, pages 5753–5763, 2019.

[104] Yan Zeng et Jian-Yun Nie : Generalized conditioned dialogue generation based on pre-trained language model, 2020.

[105] Yan Zeng et Jian-Yun Nie : Multi-domain dialogue state tracking based on state graph, 2020.

[106] Yan Zeng et Jian-Yun Nie : Open-domain dialogue generation based on pre-trained language models. *arXiv preprint arXiv:2010.12780*, 2020.

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

[108] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela et Jason Weston : Personalizing dialogue agents: I have a dog, do you have pets too? *In Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2204–2213, 2018.

[109] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu et Bill Dolan : Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.

[110] Tiancheng Zhao, Ran Zhao et Maxine Eskenazi : Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, 2017.

[111] Yinhe Zheng, Rongsheng Zhang, Xiaoxi Mao et Minlie Huang : A pre-training based personalized dialogue generation model with persona-sparse data. *arXiv preprint arXiv:1911.04700*, 2019.

[112] Victor Zhong, Caiming Xiong et Richard Socher : Global-locally self-attentive encoder for dialogue state tracking. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, 2018.

[113] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu et Bing Liu : Emotional chatting machine: Emotional conversation generation with internal and external memory. *In Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[114] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu et Xiaoyan Zhu : Commonsense knowledge aware conversation generation with graph attention. *In IJCAI*, pages 4623–4629, 2018.

[115] Li Zhou, Jianfeng Gao, Di Li et Heung-Yeung Shum : The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.

[116] Li Zhou et Kevin Small : Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*, 2019.

[117] Xianda Zhou et William Yang Wang : Mojitalk: Generating emotional responses at scale. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137, 2018.

[118] Su Zhu, Jieyu Li, Lu Chen et Kai Yu : Efficient context and schema fusion networks for multi-domain dialogue state tracking. *arXiv preprint arXiv:2004.03386*, 2020.

[119] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba et Sanja Fidler : Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *In Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.