Towards Generalized Models for Task-oriented Dialogue Modeling on Spoken Conversations

Ruijie Yan[1]*, Shuang Peng[2], Haitao Mi[2], Liang Jiang[2], Shihui Yang[2], Yuchi Zhang[2], Jiajun Li[2], Liangrui Peng[2], Yongliang Wang[2], Zujie Wen[2]

[1] Beijing National Research Center for Information Science and Technology
[1] Department of Electronic Engineering, Tsinghua University, Beijing, China
[2] Ant Group
yrj17@mails.tsinghua.edu.cn, penglr@tsinghua.edu.cn
{jianfeng.ps, haitao.mi, tianxuan.jl, yilin.ysh, yuchi.zyc, suojue.ljj, yongliang.wyl, zujie.wzj}@antgroup.com

Abstract

Building robust and general dialogue models for spoken conversations is challenging due to the gap in distributions of spoken and written data. This paper presents our approach to build generalized models for the Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations Challenge of DSTC-10. In order to mitigate the discrepancies between spoken and written text, we mainly employ extensive data augmentation strategies on written data, including artificial error injection and round-trip text-speech transformation. To train robust models for spoken conversations, we improve pre-trained language models, and apply ensemble algorithms for each sub-task. Typically, for the detection task, we fine-tune RoBERTa and ELECTRA, and run an error-fixing ensemble algorithm. For the selection task, we adopt a two-stage framework that consists of entity tracking and knowledge ranking, and propose a multi-task learning method to learn multi-level semantic information by domain classification and entity selection. For the generation task, we adopt a cross-validation data process to improve pre-trained generative language models, followed by a consensus decoding algorithm, which can add arbitrary features like relative Rouge metric, and tune associated feature weights toward Bleu directly. Our approach ranks third on the objective evaluation and second on the final official human evaluation.

1 Introduction

Although promising results have been achieved by dialogue systems on written conversations, using them directly on spoken conversations is challenging due to the differences in data distribution, including the discrepancy between writing and speaking, and the extra noises from speech recognition errors. In Dialog System Technology Challenges 10 (DSTC-10), the sub-track 2 of the "Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations" task proposes such a challenge, in which models are evaluated on spoken conversations while no spoken training data is provided. Therefore, it is crucial to build robust dialogue models with high generalization ability.

The task extends the DSTC-9 track 1 (Kim et al. 2020) from written conversations to spoken conversations, where cross-domain dialogue agents are built to answer questions that cannot be solved with only domain APIs. Instead, system agents have to retrieve related question-answer (QA) pairs from an unstructured FAQ database, and generate a natural response based on the retrieved QA pair(s). Thus, our tasks include (1) finding knowledge-seeking turns; (2) returning ranked QA pairs for each knowledge-seeking turn; and (3) generating a system response given QA pairs from task (2) and the dialogue history.

In this paper, we build generalized models for the task in the following ways. First, to bridge the gap between writing and speaking, we employ various data augmentation methods to expand the training set, including artificial error injection and round-trip text-speech transformation. Second, we use pre-trained language models (e.g. RoBERTa (Liu et al. 2019), ELECTRA (Clark et al. 2020), and UniLM (Dong et al. 2019)), and design different ensemble algorithms for each sub-task. Third, for the selection task, we propose a multi-task learning mechanism to enhance models' ability to learn multi-level semantic information. We also introduce artificial sparse features to explicitly capture informative attributes of the candidate knowledge. For the generation task, we mainly follow Mi et al. (2021), and directly use their online sampling, and consensus decoding algorithms (Pauls, DeNero, and Klein 2009).

In particular, we extend the work of Mi et al. (2021), and make the following extra contributions in this paper.

- **Data augmentation.** We augment written data by injecting artificially-generated errors based on phonetic similarity, converting the original texts into sound waves by a text-to-speech (TTS) model and then transforming back into texts by an automated speech recognition (ASR) model, and splitting or inserting entity names in dialogues.
- **Multi-task learning.** We propose a multi-task learning method for knowledge ranking, in which a domain classification task and an entity selection task are assigned to learn multi-level semantic information.
- **Incremental improvements.** We apply more pre-trained language models, e.g. ELECTRA, to increase the modeling diversity. We also introduce additional artificial sparse features to explicitly capture informative attributes of knowledge snippets.

[*] Work done during an internship at Ant Group.

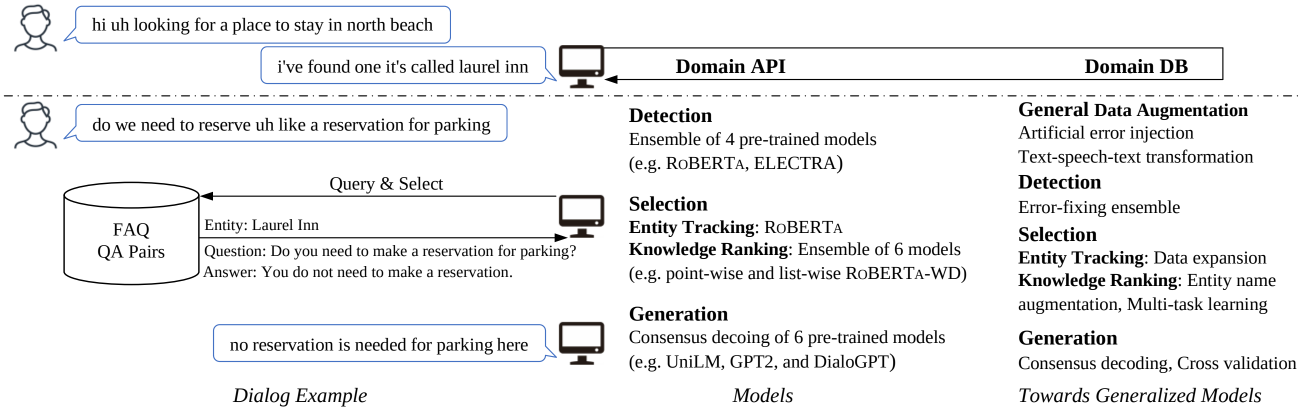
hi uh looking for a place to stay in north beach
i've found one it's called laurel inn
Domain API
Domain DB
do we need to reserve uh like a reservation for parking
Query & Select
FAQ
QA Pairs
Entity: Laurel Inn
Question: Do you need to make a reservation for parking?
Answer: You do not need to make a reservation.
no reservation is needed for parking here
Detection
Ensemble of 4 pre-trained models
(e.g. RoBERTa, ELECTRA)
Selection
Entity Tracking: RoBERTa
Knowledge Ranking: Ensemble of 6 models
(e.g. point-wise and list-wise RoBERTa-WD)
Generation
Consensus decoing of 6 pre-trained models
(e.g. UniLM, GPT2, and DialoGPT)
General Data Augmentation
Artificial error injection
Text-speech-text transformation
Detection
Error-fixing ensemble
Selection
Entity Tracking: Data expansion
Knowledge Ranking: Entity name
augmentation, Multi-task learning
Generation
Consensus decoding, Cross validation
Dialog Example
Models
Towards Generalized Models

Figure 1: An example and overview of our approach on detection, selection and generation tasks. The selection task consists of two sub-tasks: entity tracking and knowledge ranking. Entity tracking aims to find a list of entities mentioned in a dialogue, while knowledge ranking ranks a list of knowledge, and returns the top-n related knowledge.

Our system achieves the third best in the official objective evaluation, and the second best on human evaluation.

2 Task Description and Our Approach

Task-oriented dialogue systems provide information to or complete actions for users through a natural language dialogue flow. To build smart assistants that are not constrained to a set of pre-defined operations (APIs) and structural data (DBs), in the "Beyond Domain APIs" task of DSTC-9 (Kim et al. 2021), dialogue systems with high performance are proposed to utilize unstructured FAQ pairs to complete knowledge seeking tasks and generate natural responses (Mi et al. 2021; He et al. 2021; Tang et al. 2021).

While in DSTC-9 we only focused on written conversations, the "Knowledge-grounded Task-oriented Dialogue Modeling" challenge in DSTC-10 extends the task to spoken conversations. No additional spoken training data is provided. As many machine learning methods work well only under the assumption that the training and test data are drawn from the same feature space and the same distribution, the generalization ability of models trained on the DSTC-9 dataset is unsatisfactory on spoken data.

We adopt the method proposed by Mi et al. (2021) as our baseline system. Figure 1 gives a brief overview of the approach. The system first determines whether the utterance is a knowledge-seeking turn (**Detection**), then tracks entities in the dialogue history and ranks relevant QA pairs (**Selection**), and finally generates a natural response based on the retrieved QA pair (**Generation**).

We divide the selection task into two sub-tasks: **Entity Tracking** and **Knowledge Ranking**. Entity tracking aims to find a list of entities that are mentioned in dialogue. With these entities, we collect all knowledge associated with them, and knowledge ranking ranks those knowledge documents and returns top-n knowledge.

In order to build generalized models, we augment training data to simulate spoken dialogues. The current state-of-the-art pre-trained language models are employed for each task, followed by ensemble algorithms.

3 Key Components of Our Approach

3.1 Task Formulation

For simplicity, a dialogue D consists of a sequence of user and system utterances. Let U_i and S_i represent a user utterance and a system utterance at turn i respectively.

A knowledge set $\mathbf{K}$ is a list of knowledge, and each knowledge k_j is a tuple of $\langle e_j, q_j, a_j \rangle$, where e_j is an entity name or a domain name (for domain-level knowledge), q_j is a knowledge question, and a_j is a knowledge answer. We use $\mathbf{E}$ to denote a complete set of all e_j in $\mathbf{K}$. At each knowledge-seeking turn i, a ranked knowledge list K_i^R is associated with this turn. Let k_i^n represent the n-th knowledge.

Given above annotations, we define our tasks as

- **Detection** detects whether turn i is a knowledge seeking turn given the context of U_1 ... $S_{i-1}U_i$.
- **Selection** returns a top-n ranked knowledge list K_i^R at knowledge seeking turn i from the whole knowledge set $\mathbf{K}$, we divide this task into following two sub-tasks:
  - **Entity Tracking** extracts a list of entities E_i, in which each entity e_i is mentioned in the current dialogue;
  - **Knowledge Ranking** first collects a candidate knowledge list K_i^C based on entities in E_i, then ranks K_i^C and returns top-n knowledge K_i^R (k_i^1, k_i^2, ... k_i^n).
- **Generation** predicts a system response S_i^* given K_i^R and the context of U_1 ... $S_{i-1}U_i$.

Besides annotations defined above, we also add special tags to mark the type of each block in data representation and separate the input in our models:

- $\langle user \rangle$: start of a user utterance;
- $\langle sys \rangle$: start of a system utterance;
- $\langle kng \rangle$: start of a knowledge;
- $\langle kng_k \rangle$: start of the k-th best knowledge;
- $\langle ent \rangle$: start of an entity or a domain name;
- $\langle ans \rangle$: start of a knowledge answer;
- $\langle resp \rangle$: start of a response.

Sub-task	Context		Target	Type
Entity Tracking	Sentence 1	Sentence 2	True/False	Binary
	$\langle user\rangle U_1$ ... $\langle sys\rangle S_{i-1}\langle user\rangle U_i$	$\langle ent\rangle e_j$	True	
Knowledge Ranking	Sentence 1	Sentence 2	True/False	Binary
	$\langle user\rangle U_1$ ... $\langle sys\rangle S_{i-1}\langle user\rangle U_i$	$\langle kng\rangle q_j\langle ans\rangle a_j$	True	
	Sentence 1	Sentence 2	One hot	Multi-class
	$\langle user\rangle U_1$ ... $\langle sys\rangle S_{i-1}\langle user\rangle U_i$	$\langle kng\rangle q_j^1\langle ans\rangle a_j^1$, ... , $\langle kng\rangle q_j^5\langle ans\rangle a_j^5$	[0, 1, 0, 0, 0]	

Table 1: The data representations for entity tracking and knowledge ranking tasks. In the knowledge ranking task, we first use a point-wise RoBERTa to select top-5 related knowledge (rows 4 and 5), then, we develop a list-wise RoBERTa (the last two rows) to rank top-5 knowledge again, $\langle kng\rangle q_j^1\langle ans\rangle a_j^1$, ... , $\langle kng\rangle q_j^5\langle ans\rangle a_j^5$ means a batch of top-5 knowledge, and the objective function is to minimize the cross-entropy loss between the true distribution and the system prediction of five classes.

3.2 Data Augmentation

To alleviate the gap between writing and speaking, for each dialogue in the DSTC-9 dataset, we use the following three strategies to augment the utterances.

- **Artificial error injection**. We simulate speech recognition errors by randomly replacing words based on phonetic similarity (Meechan-Maddon 2019). The phonetically-similar alternatives are selected by approximate nearest neighbors search with angular distance. The proportion of replaced words is sampled from 0.1 to 0.3.
- **Text-speech-text transformation**. We synthesize sound waves from the original texts using the Tacotron 2 TTS model (Shen et al. 2018), and then adopt the Deep Speech 2 ASR model (Amodei et al. 2016) to recover texts from sound waves.
- **Entity name augmentation**. Entity name is one of the most important information for the selection task. People tend to omit certain words when they speak, and errors may occur when one or more words are omitted from the entity name. In addition, non-ground-truth entities that appeared in the conversation are easily identified as false positives. To alleviate this issue, we augment entity names in conversations according to the attributes of the candidate knowledge. On one hand, if the candidate knowledge is positive and the corresponding entity name can be matched in the dialogue, we split the entity name into two parts, and randomly move one of them to another place in the dialogue. We also delete each word in the entity name with a probability of 0.1. On the other hand, if the candidate knowledge is negative and the corresponding entity name does not appear in the conversation, we randomly insert the entity name into the dialogue.

Table 2 shows some examples for data augmentation. For entity name augmentation, we use “SW Hotel” as an example of the negative entity name. Although the augmented entity name seems unnatural in the utterance, it requires the model to understand the semantic information in the dialogue rather than simply using keywords to match knowledge. In general, errors from artificial error injection are often moderate, while text-speech-text transformation simulates more challenging situations. We apply artificial error injection and text-speech-text transformation for all three tasks on the whole DSTC-9 dataset to expand training data, while we only run entity name augmentation for the selection task with a probability of 0.3 along with the training process.

Augmentation	Text
None	can I cooking at Hamilton lodge
Error injection	can I **booking** at Hamilton **launch**
Text-speech-text	can I cooking **and high museum large**
Entity name (pos)	can I cooking **lodge** at Hamilton
Entity name (neg)	can I **SW Hotel** cooking at Hamilton lodge

Table 2: Examples of different data augmentation methods. The augmented words are marked in **bold**. For entity name augmentation, “pos” and “neg” denote the candidate knowledge is positive or negative, and we take “SW Hotel” as an example of the negative entity name.

3.3 Detection

Models and Data Representations Following Mi et al. (2021), we treat detection as a *binary* classification problem and make use of pre-trained language models, such as RoBERTa (Liu et al. 2019) and ELECTRA (Clark et al. 2020). We represent the source side as a concatenation of all dialogue utterances with tag labels, i.e. the dialogue history:

$$\langle user\rangle U_1 \text{ ... } \langle sys\rangle S_{i-1}\langle user\rangle U_i.$$

The target side is ‘True’ or ‘False’ for binary classification.

Our ensemble algorithm directly uses the error-fixing ensemble of Mi et al. (2021).

3.4 Selection

Following Mi et al. (2021), we split selection into two subtasks: entity tracking and knowledge ranking, and treat these tasks as *sentence pair* classification problems. Data representation for selection includes three parts in Table 1.

- Sentence 1: the history of a dialogue,
- Sentence 2: an entity name or a knowledge for entity tracking or knowledge ranking respectively,
- Target: the prediction space of each typical model.

Data Representations Table 1 lists data representations of our two sub-tasks. For binary models, “sentence 1” is the dialogue history, while “sentence 2” is an entity name or knowledge information, and the target is ‘True’ or ‘False’. For the multi-class or list-wise model in knowledge ranking, we put top-5 knowledge from a point-wise model in “sentence 2” in a batch, and minimize the cross-entropy loss between the true distribution and system predictions.

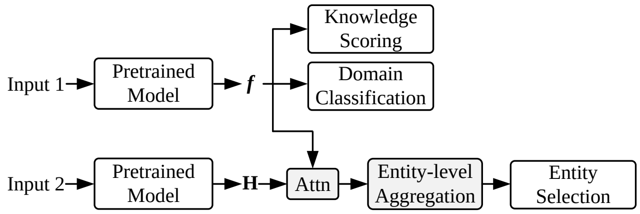

Figure 2: Illustration of the multi-task learning method for knowledge ranking. In addition to the regular knowledge scoring, we design a domain classification task and an entity selection task. Input 1 is the concatenation of Sentence 1 and Sentence 2, and Input 2 is the concatenation of several sampled entity names. $\boldsymbol{f}$ denotes the pooling output of hidden states of the encoder for Input 1, and $\mathbf{H}$ denotes the output hidden states of the encoder for Input 2.

Entity Tracking Given a complete set of entities $\mathbf{E}$ and a dialogue, entity tracking finds an entity list E_i, each entity in which is mentioned in the current dialogue. The rows 2 to 3 in Table 1 illustrate the data representation. For each entity $e_j \in \mathbf{E}$, our model predicts whether this e_j is mentioned in the current dialogue or not. In practice, e_j is mentioned if its score is larger than a threshold δ_e.

Since there are some data sets annotated in a similar way as entity tracking, we extract additional training data from Schema-Guided Dialogue (Rastogi et al. 2020), Topical-Chat (Gopalakrishnan et al. 2019) and Topical-Chat ASR (Gopalakrishnan et al. 2020).

Knowledge Ranking After we obtain an entity list E_i from entity tracking, we rank them and return a top-n knowledge list. We use two types of models: point-wise and list-wise (Cao et al. 2007). The point-wise approach scores each candidate independently and ranks them based on their scores, while the list-wise method takes into account all candidates at the same time and the objective is to maximize the probability of the correct one.

For the point-wise method, we propose a multi-task learning approach in training to learn multi-level semantic information. Typically, we design two auxiliary tasks: domain classification and entity selection in Figure 2. Input 1 and Input 2 share a same encoder. For domain classification, we use the pooling output of the last hidden states of Input 1 ($\boldsymbol{f}$), and predict the domain of the ground-truth knowledge. For entity selection, we first sample $n-1$ negative entities from $\mathbf{E}$, then insert the ground-truth entity into a random position and form Input 2 with n entity names. Our intuition is that our model should distinguish the correct entity from others according to dialogue utterances. We run the pre-trained language model over Input 1 and 2 separately, and get the last hidden states $\mathbf{H} = \{\boldsymbol{h}_i\}_{i=1:T}$ of Input 2, where T is the length of Input 2. We compute attention scores $\boldsymbol{a}$ by using $\boldsymbol{f}$ as query and $\mathbf{H}$ as keys. Then we have weighted hidden states $\mathbf{G} = \{\boldsymbol{g}_i\}_{i=1:T}$ according to Equ. (1) and Equ. (2).

$$\boldsymbol{a} = softmax(\frac{1}{\sqrt{d}}\boldsymbol{f}\mathbf{W_Q}\mathbf{W_K^T}\mathbf{H}^T) \quad (1)$$

$$\boldsymbol{g}_i = a_i\boldsymbol{h}_i\mathbf{W_V},\ i = 1, 2, ..., T \quad (2)$$

where $\boldsymbol{f} \in \mathbb{R}^{1\times d}, \mathbf{H} \in \mathbb{R}^{T\times d}$, d is the feature dimension. $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$ are projection matrices.

To get entity-level features $\mathbf{S} = \{\boldsymbol{s}_k\}_{k=1:n}$, we aggregate $\mathbf{G}$ over each entity. Let k_1 to k_t denote indices of the k-th entity in Input 2, where t is the entity length, then we have

$$\boldsymbol{s}_k = \sum_{i=k_1}^{k_t} \boldsymbol{g}_i,\ k = 1, 2, ..., n \quad (3)$$

Finally, we compute the probability distribution of n entities from $\mathbf{S}$ and minimize the KL divergence between it and the true distribution.

In practice, we sample 3 negative entities for each training instance from entities that appeared in dialogue utterances and entities that belong to the same domain as the ground-truth entity.

We further introduce two binary sparse features to indicate (1) whether the current knowledge is domain-level knowledge and (2) whether the entity name of the candidate knowledge is the last entity that appeared in dialogue utterances. We simply use string matching to check whether an entity name appear in the dialogue utterances, and we integrate sparse features into a Wide & Deep structure (Cheng et al. 2016), named RoBERTa-WD in our experiments.

For the list-wise method, we first run a point-wise model, and get top-5 knowledge for each example. Then we apply list-wise models on those top-5 results (see the last two rows in Table 1 for an example). For the training samples, we use "k-fold cross-validated style" to get system candidate lists. We split the training data into k sub-folds, and train k different models by holding each sub-set as a validation set, then decode each sub-set with the corresponding trained model.

In addition to two sparse features used in point-wise models, for list-wise models we add two extra features: whether uni-gram and bi-gram of the current entity name show in the dialogue. These features aim to help the model better deal with the challenge that the entity name is scattered in the dialogue utterances. For example, for the entity "Hilton San Francisco Union Square", the words "San Francisco" and "Union Square" may be mentioned at the beginning of the conversation, while "Hilton" may appear in the end.

Our ensemble re-ranks all knowledge by using the sum of probabilities of each knowledge from all single systems.

3.5 Generation

Following Mi et al. (2021), we make use of pre-trained language models, such as UniLM (Dong et al. 2019), GPT2 (Radford et al. 2019) and DialoGPT (Zhang et al. 2020) for our generation task, and adopt their online training and ensemble methods. For more details, please refer to Mi et al. (2021). Additionally, we perform extra data processing on DSTC-10 data in order to capture the characteristics of spoken conversations.

Data Preprocessing Besides the usage of data augmentation in Section 3.2, we also modify the response style in our training data in order to match the style of DSTC-10 validation set. Typically, the response in DSTC-9 always ends with different types of interrogative sentences, e.g. "Would

Context			Target
History	Top-5 Knowledge	Last Turn	Response
$\langle user\rangle U_1$ ... $\langle sys\rangle S_{i-1}$	$\langle kng_5\rangle\langle ent\rangle e_i^5\langle ans\rangle a_i^5$ ... $\langle kng_1\rangle\langle ent\rangle e_i^1\langle ans\rangle a_i^1$	$\langle user\rangle U_i$	$\langle resp\rangle\ S_i$

Table 3: The data representations for generation task.

System	DA	Precision	Recall	F1
RoBERTa-256	No	1.0	0.6538	0.7907
RoBERTa-256	Yes	0.9900	0.9230	0.9553
RoBERTa-512	Yes	0.9800	0.9420	0.9606
ELECTRA-256	Yes	1.0	0.9615	0.9804
ELECTRA-512	Yes	0.9717	0.9904	0.9810
Ensemble	Yes	0.9900	0.9810	0.9855

Table 4: Detection results on the validation set. "DA" means data augmentation. -256 and -512 mean the maximum context length. The RoBERTa-256 model without data augmentation is trained on the DSTC-9 training set.

you like to book a room?", which is not shown in DSTC-10 validation. Thus, we first collect some high-frequency interrogative sentences, then delete them from our training data.

Models and Data Representations Table 3 shows the data representations for generation task, where top-5 knowledge are given in a descending order from 5th to 1st. With those representations, we fine-tune UniLM (Dong et al. 2019), GPT2 (Radford et al. 2019) and DialoGPT (Zhang et al. 2020) on the training set, and pick the best models based on the scores on the validation set.

4 Experiments

4.1 Data and Common Settings

We use the DSTC-9 dataset, including training set, validation set, and test set as the original training data, then we perform data augmentation on the original training data. We evaluate our systems on the DSTC-10 validation set.

For all data representations, if a context length is larger than the maximum block size, we always trunk the left-most part of the context. "k-fold cross-validated style" means that we first split a data into k sub-folds, and train k different models by holding each sub-set as a validation set, then we decode each sub-set with the corresponding trained model.

4.2 Detection

We fine-tune pre-trained language models on our training data for 10 epochs with a learning rate of $1\text{e}{-5}$. The batch size is 16 for RoBERTa related models and 32 for ELECTRA related models. The optimizer is AdamW.

Table 4 lists results of different models on the validation set. It is clear that training on written conversations fails to perform well on spoken conversations. After we run data augmentation, we see significant improvements in terms of F1 score, and longer context length yields slightly better F1. Those results also suggest that ELECTRA models always perform better than RoBERTa in terms of F1 score. For the error-fixing ensemble, we select the ELECTRA-512 as our base model. The threshold of δ_d is 0.3, and our ensemble achieves the best F1 score at 0.9855.

System	Precision	Recall	F1
Baseline: DSTC-9	0.9017	0.7116	0.7954
Baseline: Knover	0.8967	0.6735	0.7692
Ensemble	0.8814	0.9575	0.9179

Table 5: Detection results on the test set.

System	Entity Tra.	Knowledge Ranking		
	Recall	MRR@5	R@1	R@5
Exact Match	0.8077	0.6566	0.6154	0.7115
Fuzzy Match	0.9808	0.7662	0.7115	0.8558
RoBERTa	0.9904	0.8075	0.7308	0.9231

Table 6: Comparison of different entity tracking methods on the validation set. The exact/fuzzy match methods check whether an entity name shows in dialogue utterances by exact/fuzzy matching.

Table 5 shows the results on the test set, with data augmentation and ensemble algorithms, our model achieves 0.9179 in terms of F1 score.

4.3 Selection

For the selection task, we compare different entity tracking methods, data augmentation, negative sampling strategies, and model structures. The δ_e for entity tracking threshold is 0.5. We train all models on the training data for 2 epochs with a learning rate of $1\text{e}{-5}$, a batch size of 16, and the optimizer of AdamW. For a fair comparison, we use the ground truth of detection for experiments on the validation set. As our training set includes DSTC-9 test set, and a portion of DSTC-9 test set also shows in DSTC-10 validation set but with spoken language, experimental results on the validation set are relatively high. Please also note that we add DSTC-10 validation set into our training for the final DSTC-10 test set.

Entity Tracking Table 6 shows the entity tracking and the corresponding knowledge ranking results on the validation set. The exact and fuzzy match approaches directly check whether an entity name exists in dialogue utterances by exact and fuzzy matching, respectively. Please note that the RoBERTa model is trained directly on training data with data augmentation (including more data from Schema-Guided Dialogue (Rastogi et al. 2020), Topical-Chat (Gopalakrishnan et al. 2019) and Topical-Chat ASR (Gopalakrishnan et al. 2020), and data augmentation methods in Section 3.2). The knowledge ranking model is a point-wise RoBERTa-WD without multi-task learning and trained on the original training set without data augmentation. Those results suggest that it is unsatisfactory to use exact match method due to the omission of a part of entity name and speech recognition errors, which can be alleviated by fuzzy matching. However, fuzzy matching may

Data	MRR@5	R@1	R@5
ROBERTA-WD	0.8075	0.7308	0.9231
+ AEI	0.8365	0.7596	0.9519
+ AEI + TST	0.8564	0.7981	0.9327
+ AEI + TST + ENA	0.8838	0.8173	0.9712

Table 7: The effect of data augmentation methods on the validation set. "AEI", "TST", and "ENA" denote artificial error injection, text-speech-text transformation, and entity name augmentation, respectively.

Model	MRR@5	R@1	R@5
ROBERTA-WD *	0.8822	0.8365	0.9615
ROBERTA-WD-MTL	0.8966	0.8462	0.9712
ROBERTA-WD-listwise	0.9006	0.8558	0.9712
ROBERTA-WD2-listwise	0.9290	0.9038	0.9712

Table 8: Results of different models on the validation set. ROBERTA-WD * is the point-wise model with data augmentation and better negative sampling. ROBERTA-WD-MTL adds multi-task learning. ROBERTA-WD2-listwise uses two additional sparse features to indicate whether the uni-gram and bi-gram of the current entity name appeared in the dialogue history.

introduce considerable candidate entities and bring difficulties for the subsequent knowledge ranking task. Instead, using a ROBERTA model not only achieves the highest recall rate on the entity tracking task, but also improves the performance on the knowledge ranking task.

Data Augmentation Table 7 shows the knowledge ranking results of a basic point-wise ROBERTA-WD model with different levels of data augmentation. "AEI", "TST", and "ENA" denote artificial error injection, text-speech-text transformation, and entity name augmentation, respectively. By adding more and more data augmentation methods, we see consistent and significant improvements in terms of R@1. Although text-speech-text transformation degenerates the R@5 score, it boosts R@1 and MRR@5 significantly. Together with all three data augmentation methods, we improve MRR@5 from 0.8075 to 0.8838, R@1 from 0.7308 to 0.8173, and R@5 from 0.9231 to 0.9712.

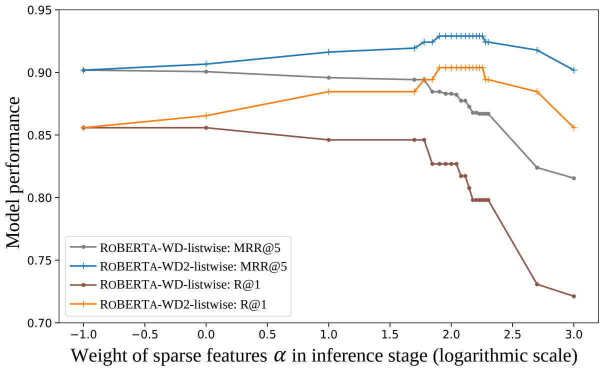

Figure 3: The influence of the weight of sparse features in inference stage on the performance for list-wise models. We only show R@1 and MRR@5 because R@5 remains unchanged for all list-wise models.

Model	MRR@5	R@1	R@5
Baseline: DSTC-9	0.5230	0.4583	0.6252
Baseline: Knover	0.5574	0.4950	0.6472
ROBERTA-WD2-listwise	0.7541	0.7004	0.8225
Ensemble	0.7891	0.7481	0.8407

Table 9: Selection results on the test set. ROBERTA-WD2-listwise is our best single model trained on the training set, and our ensemble model further contains models trained on both training set and DSTC-10 validation set.

Negative Sampling In the original knowledge ranking system of Mi et al. (2021), the negative training instances for point-wise knowledge ranking models are equally sampled from (1) the complete knowledge base and (2) knowledge about entities that appeared in dialogue utterances. However, we found that the knowledge of non-ground-truth entities in the conversation is more likely to be recognized as a false positive candidate than knowledge of a ground-truth entity. Thus, we extend their negative sample candidates by adding a knowledge from a non-ground-truth entity that also shows in a conversation. This negative sampling strategy significantly improves R@1 from 0.8173 further to 0.8365.

Comparison of different models Table 8 lists the results of different models on the validation set. For point-wise models, with the help of multi-task learning method, our ROBERTA-WD-MTL improves all metrics by about 1 percent. For list-wise models, ROBERTA-WD-listwise system uses two binary sparse features to indicate the domain of the current knowledge and whether the current entity is the last entity in the conversation. The list-wise ranking helps about another 1 percent in terms of R@1. The ROBERTA-WD2-listwise system further adds two additional sparse features to identify whether the uni-gram and bi-gram of the current entity name appeared in the dialogue history, and improves the R@1 by almost 5%.

In order to better utilize sparse features, we scale sparse feature indicator by a weight α, which means that the input of a fired sparse feature is α instead of 1. Figure 3 shows the performance curves with different α on the validation set. It is interesting that as α goes up, ROBERTA-WD-listwise gets worse, while ROBERTA-WD2-listwise performs better before reaching a threshold. It suggests that increasing unigram and bi-gram feature weights to a range from 80 to 180 leads to the best performance on the validation set. Thus, we fix α to be 100 in inference stage, our best single model achieves 0.9038 in terms of R@1 (the last row in Table 8).

Final Results on Test Table 9 presents results of our best single model (ROBERTA-WD2-listwise) and ensemble model on the final test set. ROBERTA-WD2-listwise is trained on the training set, and our ensemble model further contains models trained on both training set and DSTC-10 validation set. The proposed techniques, including data augmentation, multi-task learning, artificial sparse features, and ensemble algorithm bring significant performance gains over baseline systems.

Type	System	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
Single	UniLM Uncased	0.1509	0.0703	0.0101	0.0125	0.1532	0.2007	0.0470	0.1188
	UniLM Large	0.1909	0.1103	0.0501	0.0204	0.2140	0.2501	0.0970	0.1689
	GPT2	0.1628	0.0966	0.0544	0.0271	0.1839	0.2092	0.0859	0.1406
	DialoGPT	0.1544	0.0813	0.0452	0.0169	0.1655	0.1840	0.0721	0.1396
Ensemble	10 features	0.2072	0.1435	0.0980	0.0560	0.1834	0.2501	0.1334	0.2172
	+ DA	0.3479	0.2512	0.1658	0.1260	0.3885	0.3735	0.2061	0.3267
	+ DA & DP	0.3613	0.2932	0.2405	0.1871	0.4295	0.4170	0.2816	0.3670

Table 10: Validation results of top four single systems (ranked by BLEU-4), and consensus decoding with 10 basic features and data augmentation. 10 basic features include 9 similarity features, and 1 reciprocal rank feature (Mi et al. 2021). "DA" means data augmentation methods shown in Section 3.2, and "DP" means data preprocessing of removing high frequent interrogative sentences from responses.

Type	Sys	BLEU-1	BLEU-4
Baseline: DSTC-9		0.1153	0.0075
Baseline: Knover		0.1248	0.0153
Single	UniLM Large	0.1802	0.0460
Ensemble	10 features	0.3150	0.0732
	+ DA	0.3228	0.1019
	+ DA & DP	0.3401	0.1154

Table 11: Test results of our models. We only show BLEU-1 and BLEU-4 due to space limitation.

System	Accuracy	Appropriateness	Average
Ground-truth	3.5769	3.4814	3.5292
Best	3.4947	3.3523	3.4235
Ours	3.3356	3.3021	3.3189

Table 12: Human evaluation results on the test set.

4.4 Generation

We fine-tune pre-trained models with batch size 32 for 6 epochs over the training data. We use three UniLM models[*]. For the UniLM Large models, the maximum history size of each context is 512, and the maximum target size is 96. The optimizer is AdamW with a learning rate of 1e−5, weight decay of 0.01. UniLM Uncased models use 640 as the maximum history size. The p_s of UniLM Large is 0.15. GPT2 and DialoGPT models use a learning rate of 5e−5, and the maximum block size is 512. We use base models for GPT2 and DialoGPT.

Discrepancy in Pipeline Framework Following Mi et al. (2021), we decode the training set in "*10*-fold cross-validated style" to alleviate the discrepancy in the pipeline framework. The R@1 score of the decoded training set is about 0.9. We then train our model on the decoded training set, and observe 1 percent improvement in terms of BLEU-4.

Results Table 10 shows top four single systems and ensemble results on the validation set. UniLM Large uses top-5 knowledge in context, and performs best on 6 of 8 metrics. UniLM Uncased achieves comparable results to UniLM Large. Please note that, all the single systems shown in Table 10 only use DSTC 9 training set without any data augmentation. Our consensus decoding with 10 basic features improves BLEU-4 scores by 3.6 percent on the validation set, After we perform data augmentation in Section 3.2, the BLEU-4 scores are improved by 7 percent, and the data preprocessing by removing high frequent interrogative sentences further improves BLEU-4 scores by 6.1 percent.

Table 11 shows the results on the final test set. Our single model UniLM Large is significantly better than official baseline systems, and consensus decoding improves BLEU-4 by 2.7 points. Adding DA and DP further improve BLEU-4 scores by 2.8 and 1.4 percent, respectively. Those improvements are smaller than the improvements on our validation set due to the following two facts: 1) the test set is about 10 times larger than validation set; 2) the validation set includes partial DSTC 9 test set, which is used in our training.

4.5 Human Evaluation Results

Table 12 shows the official human evaluation results on the test set, 'appropriateness' measures how well a system output is naturally connected to a given conversation, while 'accuracy' measures the accuracy of a system output against the reference knowledge. Our ensemble system ranks in the second place.

5 Conclusion

In this paper, we have presented our system pipeline for the sub-track 2 of "Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations" in DSTC-10. To bridge the gap between speaking and writing, we adopt extensive data augmentation methods. We also use multi-task learning, and different ensemble algorithms for different sub-tasks to train robust models with high generalization ability. Overall, experimental results show that data augmentation significantly boosts model performance on all three sub-tasks, the multi-task learning enhances the model's ability to learn multi-level semantic information on the knowledge ranking task, and all ensemble algorithms improve the final objective metrics. Our approach has ranked third on objective metrics and second on human evaluation. In future work, we will further improve the generalization ability of our models by generating more spoken-like training data and increasing model diversity.

[*] https://github.com/microsoft/unilm/tree/master/s2s-ft

References

Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; et al. 2016. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *ICML*.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, 129–136. New York, NY, USA: ACM. ISBN 978-1-59593-793-3.

Cheng, H.-T.; Koc, L.; Harmsen, J.; et al. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS 2016*, 7–10.

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. *CoRR*, abs/1905.03197.

Gopalakrishnan, K.; Hedayatnia, B.; Chen, Q.; Gottardi, A.; Kwatra, S.; Venkatesh, A.; Gabriel, R.; and Hakkani-Tür, D. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, 1891–1895.

Gopalakrishnan, K.; Hedayatnia, B.; Wang, L.; Liu, Y.; and Hakkani-Tür, D. 2020. Are Neural Open-Domain Dialog Systems Robust to Speech Recognition Errors in the Dialog History? An Empirical Study. In *INTERSPEECH*.

He, H.; Lu, H.; Bao, S.; Wang, F.; Wu, H.; Niu, Z.; and Wang, H. 2021. Learning to Select External Knowledge with Multi-scale Negative Sampling. *AAAI DSTC9 Workshop*.

Kim, S.; Eric, M.; Gopalakrishnan, K.; Hedayatnia, B.; Liu, Y.; and Hakkani-Tur, D. 2020. Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access. arXiv:2006.03533.

Kim, S.; Eric, M.; Hedayatnia, B.; Gopalakrishnan, K.; Liu, Y.; Huang, C.-W.; and Hakkani-Tur, D. 2021. Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access Track in DSTC9. *arXiv preprint arXiv:2101.09276*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Meechan-Maddon, A. 2019. The effect of noise in the training of convolutional neural networks for text summarisation.

Mi, H.; Ren, Q.; Dai, Y.; He, Y.; Sun, J.; Li, Y.; Zheng, J.; and Xu, P. 2021. Towards Generalized Models for Beyond Domain API Task-oriented Dialogue. *AAAI DSTC9 Workshop*.

Pauls, A.; DeNero, J.; and Klein, D. 2009. Consensus Training for Consensus Decoding in Machine Translation. In *EMNLP*, 1418–1427.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Rastogi, A.; Zang, X.; Sunkara, S.; Gupta, R.; and Khaitan, P. 2020. Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset. arXiv:1909.05855.

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerry-Ryan, R.; Saurous, R. A.; Agiomyrgiannakis, Y.; and Yonghui, W. 2018. Natural TTS Synthesis By Conditioning Wavenet On Mel Spectrogram Predictions. In *ICASSP*.

Tang, L.; Shang, Q.; Lv, K.; Fu, Z.; Zhang, S.; Huang, C.; and Zhang, Z. 2021. RADGE: Relevance Learning and Generation Evaluating Method for Task-oriented Conversational System. *AAAI DSTC9 Workshop*.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2020. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL, system demonstration*.