A Difference-aware Ensemble Method for Task-oriented Dialogue with Subjective Knowledge

Changxin Ke[*] and **Churui Sun**[*] and **Longxuan Ma**[*] and **Weinan Zhang**[†] and **Ting Liu**
Research Center for Social Computing and Information Retrieval
Faculty of Computing, Harbin Institute of Technology
{cxke,crsun}@stu.hit.edu.cn {lxma,wnzhang,tliu}@ir.hit.edu.cn

Abstract

We participate in the 11th Dialog System Technology Challenges (DSTC) track-5[1] called Task-oriented Conversational Modeling with Subjective Knowledge. Introducing subjective knowledge into task-oriented dialogue (TOD) can help the DS to understand variables of subjective user needs and to suit more dialogue scenarios. Track-5 includes several sub-tasks: 1) knowledge-seeking turn detection; 2) knowledge entity tracking; 3) knowledge entry selection; and 4) use of the selected knowledge entries for response generation. Besides the challenges of each sub-tasks own, there are two challenges across different sub-tasks. The first is that there are multiple valid knowledge entries for each knowledge-seeking turn, the accuracy of the knowledge entry selection is important for the quality of response generation. The second challenge is how to address the unseen dialogue/entities/entries in the validation and the test set. In this paper, we propose a difference-aware ensemble method to address these sub-tasks and the two challenges mentioned above. Our method helps to obtain more robust results and performs well on unseen instances. Among all the submissions for the test set, our method ranks 1st on the knowledge-seeking turn detection task and achieves 3rd on the overall automatic evaluation score. Our code and data will be released on GitHub.

Subjective Knowledge (the first review):
"0": "My work colleagues and I recently visited Nandos City Centre for lunch."
"1": "We tried many Portuguese dishes such as Alheira, Caldo Verde, Francesinha, and Torricado and washed everything down with beer."
"2": "The dishes were cooked perfectly and we were able to order a lot as the prices for the dishes and the beer were very reasonable."
"3": "It did take a little longer for the food to arrive at the table. Our plates took a while to clear from the wait staff."
"4": "But overall it was a good lunch."

Subjective Knowledge (the second review):
"0": "Yesterday me and some coworkers went to Nando's City Centre."
"1": "I've never had Portuguese cuisine but my coworker insisted on it."
"2": "I was disappointed with how expensive everything was though and I hoped everything tasted good."
"3": "I was surprised with how much food there was overall, the portions were quite generous."
"4": "I'm not sure I liked the food, it didn't live up to the hype for me."

Dialogue:
User: What do you know about Nandos City Centre?
Agent: Which part do you prefer to know?
User: How is the price? Is it expensive?
Agent: The price is very reasonable because the portions were quite generous.
User: Do we need to wait a long time for the food?
Agent: Yes, some customers say it did take a little while for the food to arrive.

Table 1: SK-TOD example that the agent produces response based on the subjective knowledge (two reviews are both under the entity "Nandos City Centre").

1 Introduction

Task-oriented Dialogue Systems (TODS) aim to assist users in domain-specific tasks, such as booking a traveling ticket or a restaurant. The TODS is usually based on a domain-specific API or Database (DB) that can provide the required information for accomplishing the task, such as the price of a ticket or the location of a restaurant. However, these APIs or DBs usually do not contain information for subjective user requests (e.g., "Is this a good place for meeting colleagues?" or "Does the hotel have a good atmosphere?"). TODS trained with these APIs/DBs can only handle a limited range of scenarios. To address this issue, Zhao et al. (2023) propose a novel task called subjective-knowledge-based TOD (SK-TOD). One SK-TOD example is shown in Table 1, where the dialogue agent (DS) needs to use subjective knowledge to complete a dialogue. According to Zhao et al. (2023), a TODS should consider multiple reviews with both positive

[*] These three authors contributed equally.
[†] Corresponding author
[1] https://github.com/alexa/dstc11-track5

Proceedings of The Eleventh Dialog System Technology Challenge, pages 216–225
September 11, 2023 ©2023 Association for Computational Linguistics

and negative opinions, along with their respective proportions to make a response (as exemplified in Table 1). This two-sided response is recognized as more credible and valuable for customers, thereby fostering trust in the TODS. The SK-TOD data is used as the track-5 of the 11th Dialog System Technology Challenges (DSTC 11) workshop[2].

The track-5 (called Task-oriented Conversational Modeling with Subjective Knowledge) challenge can be divided into four sub-tasks (Zhao et al., 2023): 1) knowledge-seeking turn detection. It takes the dialogue context as input and decides whether the response to this context requires selecting external knowledge; 2) knowledge entity tracking. It selects entities from the entity pool of the subjective knowledge; 3) knowledge entry selection. It selects knowledge entries related to the selected entity from the subjective knowledge; 4) response generation uses the selected knowledge entry for generating a response. The knowledge entity tracking and entry selection sub-tasks are performed only when a response turn is detected as a knowledge-seeking turn. Each of the four sub-tasks has its own challenge. However, there are two main challenges in this competition that across different sub-tasks. The first is that there are multiple related subjective knowledge entries for each knowledge-seeking turn. This affects all knowledge entity tracking, knowledge entry selection, and response generation. The second is that there is a big portion of unseen instances in the validation and test set (see Table 3), which includes unseen dialogues/entities/entries in the train set. This affects all sub-tasks.

In this paper, we propose a difference-aware ensemble method to address the two challenges across sub-tasks. For each sub-task, we start by training a group of models that are good at specific abilities (e.g. unseen instances, noisy environment). Then we aim to combine the advantages of each model for different sub-tasks and for unseen instances. In other words, we need to assign a reasonable weight for each model based on the desired metric. To this end, we need to jointly consider three differences: 1) the different requirements of each sub-task; 2) the difference between each trained model; 3) the difference between the validation set and the test set. Our ensemble method can automatically balance these differences among tasks/models/datasets and use a unified pattern to cope with different task scenarios. Experiments show that our method largely improves performance. Among all the submissions for the DSTC 11 track-5 test set, our method ranks 1st on the knowledge-seeking turn detection task and achieves 3rd on the overall automatic evaluation score. To sum up, our contributions are:

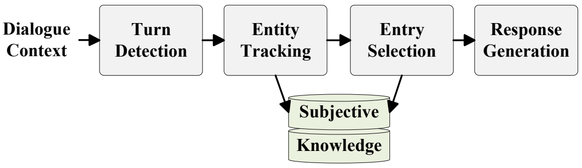

Figure 1: The pipeline we used for the SKTOD task.

- To address the multiple knowledge entries and the unseen instance problems, we propose a difference-aware ensemble method to leverage the advantages of different models that are good at specific abilities. Our method can automatically balance the differences between tasks/datasets/models.
- Experiments show that our method outperforms strong baselines and performs well on the DSTC 11 track-5. We give a detailed analysis of the experiments. Our code and data will be released on GitHub.

2 Related Work

We introduce related work in this section, including knowledge-grounded conversation, task-oriented dialogue, large language models, and ensemble methods.

2.1 Knowledge-grounded Conversation

Knowledge-grounded conversation task first choose context-related knowledge from external sources (can be structured knowledge graph or unstructured text) and use the selected knowledge to construct a response (Zhou et al., 2018; Moghe et al., 2018; Dinan et al., 2019; Gopalakrishnan et al., 2019; Jang et al., 2022; Zhao et al., 2022). The SK-TOD task is similar to KGC with unstructured text as external knowledge, especially the knowledge selection (Kim et al., 2020a; Meng et al., 2020; Xu et al., 2022) and response generation (Zhao et al., 2020; Prabhumoye et al., 2021; Majumder et al., 2022) sub-tasks. However, the KGC task usually does not contain

[2] There are also a set of FAQs besides the subjective review knowledge. However, the dialogue is mainly grounded on subjective knowledge. Please refer to Table 3 for more details.

knowledge-seeking turn detection task and are usually open-domain dialogue.

2.2 Task-oriented dialogue and Recommendation

Task-oriented dialogues (TOD) usually rely on domain-specific APIs and databases to support the dialogue response (Levin et al., 2000; Zhao et al., 2017; Akasaki and Kaji, 2017; Yan et al., 2017). Later works ground task-oriented dialogues to web pages (Chen et al., 2022), government service documents (Feng et al., 2020), and FAQ knowledge snippets (Kim et al., 2020b). Different from these works where factual knowledge is utilized, we apply subjective knowledge to generate the response and ground in multiple knowledge snippets. There are also work in TOD (Majumder et al., 2022) and recommendation (Ni et al., 2019) that aim to generate review-based response or recommendation explanations. However, the SK-TOD requires grounding in multiple subjective knowledge and explicitly considers the diversity of opinions and the proportion of sentiments.

2.3 Ensemble Methods

The traditional ensemble iteratively minimizes the loss of the joint output (Hastie et al., 2001) and keeps the child model unchanged. Another ensemble pattern is pseudo-ensemble where the child models share parameters and structure through their parent model, which will tend to correlate the child models' behavior (Hinton et al., 2012; Bachman et al., 2014). The ensemble method is usually used in DSTC competitions (Tan et al., 2020; Chaudhary et al., 2021). Our methods belong to traditional ensemble methods that learn a robust model to input uncertainty. The difference is that our difference-aware ensemble method can jointly adjust different models/datasets/tasks.

2.4 Large Language Models

Large language models (LLMs) such as GPT-4 (OpenAI, 2023) and LLAMA (Touvron et al., 2023) show that pre-trained models with a large amount of parameters (usually more than 10 Billion) can understand complex instructions and perform well in a variety of tasks including dialogue. However, despite their success, recent studies (Bubeck et al., 2023) show that LLM still has the hallucination problem and could not perform well in domain-specific or knowledge-intense tasks even with elaborately designed instructions. In this paper, we test the response generation task with LLAMA-13B. The input is dialogue context and the selected knowledge entries. However, its performance is not as good as a smaller model such as the BART baseline.

3 Our Method

In this section, we first introduce how we train the models for each sub-tasks in DSTC 11 track-5, then introduce the difference-aware ensemble method.

3.1 Problem Statement

Formally, we have a dialogue context $\mathbf{C}$ = $[U_1, A_1, U_2, A_2, ..., U_{|C|}]$ where each user turn U_i (i ∈ [1,$|C|$]) is followed by a agent response turn S_i except the last user turn $U_{|C|}$. The dialogue involves one or multiple entities represented as $\mathbf{E}^C$ = $\{E_1^C, E_2^C, ..., E_{|E^C|}^C\}$. Each entity E_i^C (i ∈ [1,$|E^C|$]) has a group of subjective knowledge entries $\mathbf{K}^{E_i^C}$ = $[K_1, K_2, ..., K_{|K^{E_i^C}|}]$[3]. The entity and their corresponding subjective knowledge are from a subjective knowledge source $\mathbf{B}$ = {($\mathbf{E}_1$, $\mathbf{K}^{E_1}$), ($\mathbf{E}_2$, $\mathbf{K}^{E_2}$), ..., ($\mathbf{E}_{|B|}$, $\mathbf{K}^{E_{|B|}}$) }. $|C|$, $|E^C|$, $|K^{E_i^C}|$, and $|B|$ are the number of elements in the corresponding set. The DSTC 11 track-5 needs first identify whether $U_{|C|}$ has a knowledge-seeking request and, if yes, then identifies a group of entities and their corresponding knowledge entries from the knowledge source $\mathbf{B}$, then selects the most relevant subjective knowledge entries from the group, finally generates a response $A_{|C|}$ grounded on the selected entries. The pipeline for SK-TOD is illustrated in Figure 1. Next, we will introduce how we train models for each sub-tasks.

3.2 Knowledge-Seeking Turn Detection (KTD)

The KTD is a binary classification problem to identify whether the last user turn in context $\mathbf{C}$ requires to be addressed with subjective knowledge. The baseline is an auto-encoding pre-trained language model (DeBERTa-v3-base (He et al., 2021)). The baseline takes dialogue context $\mathbf{C}$ as input and uses the hidden state of the first token as its representation. Then it applies a classifier as follows to obtain the probability $P(\mathbf{C})$ that the current user request is a subjective knowledge-seeking turn:

[3] Each K_j can be paragraph, sentence, or segment.

$$h_C = \text{Enc}(\mathbf{C}), \tag{1}$$

$$P(\mathbf{C}) = \text{softmax}(\text{FFN}(h_C)). \tag{2}$$

The model is fine-tuned with the binary cross-entropy loss. As we introduced in Section 1, we want to obtain a group of models that are good at specific abilities (e.g. unseen instances, noisy environment) and then combine the advantages of each model for different sub-tasks. For the KTD task, we train three models.

We define the first model as an expert on seen dialogue data. It is similar to the baseline model given by the DSTC 11 track-5 organizers. We further fine-tune the model with the training set and select the checkpoint with the best performance on the validation set.

We define the second model as an expert on unseen dialogue data. To this end, we construct an unseen dialogue dataset with the given subjective knowledge and use this new unseen dataset to fine-tune a RoBERTa (Liu et al., 2019) model[4]. The unseen dialogue dataset combines original training dialogue data and the external knowledge Tan et al. (2020). This competition's training dialogue data usually starts with talking about an entity. For each entity, there are question-answering (QA) pairs in the corresponding knowledge source **B**. We first keep the starting 2 turns of a training example and randomly select the QA pairs related to this entity in **B**. We concatenate the selected QA pairs to the starting turns. Then we obtain a new unseen dialogue example that focuses on this entity. To mimic the situation of a topic shift in the real scenario, we check whether the last turn of the new example contains another entity in **B**. If so, we continue selecting QA pairs related to the second entity and add the QA pairs to the dialogue. Finally, we obtain a dialogue example that does not appear in the original training data. In practice, we use at most three entities and restrict the total length of the dialogue to 10 turns (which is similar to the average length of the training data). The length of turns under each entity is randomly decided. Training an unseen expert with this data can help the performance of unseen dialogue instances.

We define the third model as a de-noise expert. To this end, we pre-trained and fine-tuned a DeBERTa (He et al., 2021) model with an enlarged noisy training set so that the model is more aware of the noisy inputs. The pre-training is a classical word-masking training (Devlin et al., 2019; Liu et al., 2019) that randomly masks a portion (15%) of input words and asks the language model to recover them. The enlarged noisy data is obtained with a back-translation and synonym substitution process. We adopt the Google translation service[5] to translate English into other languages (such as Spanish/German/Japanese/French), then back-translated them into English[6]. Finally, we obtain 5 times dialogue and knowledge data. These data are used to pre-train the language models. We further pair the 5-times dialogue data with knowledge translated from different languages, which gives 25 times data for fine-tuning. These data are considered noise because the back-translation and synonym substitution introduces word-level and semantic-level disturbance.

After getting the seen expert, unseen expert, and de-noise expert models. We use a difference-aware ensemble method to combine their advantages, which will be introduced in Section 3.6.

3.3 Knowledge Entity Tracking (KET)

The goal of KET is to identify the entities that are relevant to the last user turn request. It can help to reduce the number of candidates in the step of knowledge selection. The baseline model given by the DSTC 11 track-5 organizers is a word-matching-based method (Jin et al., 2021) to extract relevant entities. It first normalizes entity names in the knowledge source using a set of heuristic rules. Then a fuzzy n-gram matching is performed between the normalized entity and all dialogue turns. To find the entities that are relevant to the last user request, the baseline method chooses the last dialogue turn in which the entities are detected and uses these entities as the output. We follow the baselines and have Precision=0.9516, Recall=0.9841, F1=0.9676, and accuracy=0.9398 on the validation set. It should be noted that the competition did not require to calculate results for the KET task. Meanwhile, we are limited by the schedule of the competition and fail to train well-performed experts on this sub-task. However, we believe our ensemble method is also useful for this task and we leave this training for future work.

[4] We use different PLMs because we want the difference between the seen and unseen experts to be more pronounced

[5] https://translate.google.com

[6] When the back-translation sentence is the same as the original sentence, we employ synonym substitution with Wordnet (https://wordnet.princeton.edu/) to increase diversity.

3.4 Knowledge Entry Selection (KES)

The goal of KS is to select the knowledge entries that are relevant to the user's request. The inputs are the dialogue context **C** and the knowledge candidates **K**, which is a combination of all the knowledge entries of the relevant entities in $\mathbf{E}^C$. The output $\mathbf{K}^+$ is a subset of **K** with the most relevant knowledge candidates. Note that there might be multiple knowledge snippets in $\mathbf{K}^+$. To select relevant knowledge snippets, the baseline proposed by DSTC organizers uses a bi-encoder (Mazaré et al., 2018) structure to calculate the relevance score between the dialogue context **C** and a knowledge entry $K_i \in \mathbf{K}$. The **C** and K_i are encoded separately using the same pre-trained encoder and obtain two representations, h_C and h_{K_i}. Then the concatenation of h_C, h_{K_i}, and $|h_C - h_{K_i}|$ is used as features to obtain the probability of relevance $P(\mathbf{C}, K_i)$ as follows:

$$h_C = \text{Enc}(\mathbf{C}), h_{K_i} = \text{Enc}(K_i), \tag{3}$$

$$h_{feature} = [h_C; h_{K_i}; |h_C - h_{K_i}|], \tag{4}$$

$$P(\mathbf{C}, K_i) = \text{softmax}(\text{FFN}(h_{feature})). \tag{5}$$

Similar to the KTD task, we also train three different models for KES. There are also denoted the seen expert, the unseen expert, and the de-noise expert.

3.5 Response Generation (RG)

The goal of RG is to use dialogue context **C** and the selected knowledge entries $\mathbf{K}^+$ to construct a response. The baseline for RG is a pre-trained generation model (BART (Lewis et al., 2020)) that concatenates **C** and $\mathbf{K}^+$ as the input and generates the response. The model is trained to maximize the generation probability P(A_t | **C**, $\mathbf{K}^+$). We fine-tuned BART and T5 (Raffel et al., 2020) for this task.

3.6 Ensemble Method

Algorithm 1 shows how our ensemble method balances the advantages of different models. Taking the knowledge entry selection for example, we calculate top N candidates for the k-th validation example from each model and sort them in descending order with respect to model confidence[7]. The S_k^{gt} is ground-truth results for this example with K entries, K < N. Each candidate is given a weight which is the reciprocal of its ranking number plus one. For instance, candidates from DeBERTa (seen expert) are S_j^D, ($j = 1, 2, ..., N$), and the corresponding weight is $W_j^D = \frac{1}{j+1}$. Similarly, S_j^R and W_j^R for RoBERTa (unseen expert), S_j^E and W_j^E for DeBERTa (de-noise expert). Then we use these candidates to form a final candidate dictionary S = $\{S_i$, $(i = 1, 2, ..., T)\}$, $N \leq T \leq 3N$. The ensemble weight W_i of S_i, is calculated by W_i = $p^D \cdot \tilde{W}_i^D$+$p^R \cdot \tilde{W}_i^R$+$p^E \cdot \tilde{W}_i^E$, $(i = 1, 2, ..., T)$, p is the hyper-parameter that we want to learn, p^D+p^R+p^E = 1. $\tilde{W}_i^D$ = W_j^D if there is a j such that $S_j^D \cong S_i$, 0 otherwise. $\cong$ means exact match here. $\tilde{W}_i^R$ and $\tilde{W}_i^E$ follows the same definition as $\tilde{W}_i^D$. Then we use a specific **Metric**, such as Recall/Precision/F1/EM (or the combination of them), to learn the optimal $\hat{p}$ with all examples in the validation set. We set a threshold for S and only reserve the candidates with W_i larger than the threshold, denoted by $S_{threshold}$. The threshold is set to $\frac{1}{6}$ according to the average entry numbers in Table 3. During testing, we use the threshold to select a set of candidates from S as our final prediction using the learned weight[8].

Algorithm 1: Difference-aware ensemble method.

```
During training:
Input: $S^D$, $S^R$, $S^E$, $S$, $\tilde{W}^D$, $\tilde{W}^R$, $\tilde{W}^E$, $S_{gt}$.
Output: Weight for each model.
for $p \in$ range(start=0, stop=1, step=0.1) do
  Score = 0
  for $k \in$ {validation set} do
    Initialize W: $\{W_i = 0, i = 1, 2, ..., T\}$
    for i $\in$ [1, T]; do
      $W_i$ = $p^D \cdot \tilde{W}_i^D$+$p^R \cdot \tilde{W}_i^R$+$p^E \cdot \tilde{W}_i^E$
    end for
    Score += Metric($S_k^{threshold}$, $S_k^{gt}$)
  end for
  Record weight $\hat{p}$ for the Best Score.
end for
During test:
for $k \in$ {test set} do
  Initialize W: $\{W_i = 0, i = 1, 2, ..., T\}$
  for i $\in$ [1, T]; do
    $W_i$ = $\hat{p}^D \cdot \tilde{W}_i^D$ + $\hat{p}^R \cdot \tilde{W}_i^R$ + $\hat{p}^E \cdot \tilde{W}_i^E$
  end for
  $S_k$ = $S_k^{threshold}$
end for
```

Next, we explain why this method can balance different tasks. According to the experiments of Zhao et al. (2023), different sub-tasks in SK-

[7] In this task, the confidence is a score between 0 and 1 since we train the turn detection and entry selection with binary classification loss.

[8] For example, an entry ranks 1st in DeBERTa (seen), 3rd in RoBERTa (unseen) and 4th in DeBERTa (de-noise), p^D=0.2, p^R=0.3, p^E = 0.5 then the final weight to re-rank this entry in S is 0.2*0.5 + 0.3*0.25+0.5*0.2 = 0.275.

Models	On validation set			On final-test set		
	Precision	Recall	F1	Precision	Recall	F1
DeBERTa-v3-base (baseline)	**1.0000**	0.9990	0.9995	**0.9982**	0.9979	0.9980
DeBERTa-v3-base (seen expert)	0.9979	0.9989	0.9982	-	-	-
RoBERTa-large (unseen expert)	0.9953	0.9946	0.9949	-	-	-
DeBERTa-v3-large (de-noise expert)	0.9960	0.9983	0.9980	-	-	-
Ensemble with Precision	**1.0000**	0.9990	0.9995	-	-	-
Ensemble with Recall	0.9990	**1.0000**	0.9995	0.9979	0.9989	0.9984
Ensemble with F1	**1.0000**	**1.0000**	**1.0000**	0.9979	**0.9993**	**0.9986**

Table 2: Experimental results on the knowledge-seeking turn detection.

TOD requires different metric. For example, the knowledge-seeking turn-detection task can be seen as the simplified version the Algorithm 1. The N candidate is reduced to 1 in the turn detection task, i.e. the S_k^{gt} only has 1 ground-truth result (yes or no). We can use Exact Match between S and S_k^{gt} to learn the weight $\hat{p}$. For the KES task, both precision and recall matter since there can be multiple knowledge entries for each knowledge-seeking turn. Instead of selecting the top few results with the threshold, we can also set a threshold for the relevance score in Section 3.4.

At last, the difference-aware method is also suitable for different datasets. For example, the test set contains more portion unseen instances and multiple entity instances than the validation set. We can adjust the portion of unseen instances of the validation set so that it is similar to that of the test set. Then we can use Algorithm 1 to learn a $\hat{p}$ more suitable for the test set.

	Train	Val	Test
dialogue instances	14,768	2,129	2,798
seen instances	14,768	1,057	1,349
unseen instances	0	1,072	1,449
multi-entity instances	412	199	436
Knowledge entries			
Avg.entries / instance	3.80	4.07	4.21
Avg.review / instance	3.51	3.94	3.91
Avg.FAQ / instance	0.29	0.23	0.29
Avg.tokens / entry	15.14	15.81	14.92
Dialogue			
Avg.uttrances / instance	9.29	9.44	9.36
Avg.tokens / request	8.66	8.95	9.13
Avg.tokens / response	24.27	23.69	24.05

Table 3: Data statistics of the DSTC 11 track-5.

4 Experimental Settings

4.1 Data

The data set is provided by the organizers of the track[9]. The statistics are shown in Table 3. We can see that there are around 50% unseen instances in validation and test sets. The portion of the instances requiring multiple entities is also increased in the validation and test sets.

4.2 Evaluation Metrics

We use the metrics provided by the DSTC organizers [10] to show the automatic results. For the knowledge-seeking turn detection task, we report the Precision, Recall, and F1 scores. For entity tracking, we report the instance-level Accuracy score. An instance is regarded as accurate only if the predicted entities are the same as the gold entities. For knowledge entry selection, we report Precision, Recall, F1, and Exact Match (EM). For response generation, we report BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-1/2/L (Lin, 2004). In the competition, the KTD task servers as the foundation of the rest sub-tasks. Specifically, when calculating the metric for each sub-task, the true positive score of sub-task 1 serves as a weight and is multiplied by other results. The overall score is calculated by the mean reciprocal rank over all the metrics.

4.3 Implementations

Our implementations of RoBERT, DeBERTa, BART, and T5 are based on the public Pytorch implementation of Transformers[11]. During pre-training, we follow the hyper-parameters setting of the original implementation. During pre-training and fine-tuning, we set the maximum input length to 512 tokens. Candidate size N in Algorithm 1 is set to 10. We use a single Tesla v100s GPU with 32 GB memory for the experiments, the pre-training time is around 48 hours and fine-tuning time is around 4 hours for each model.

5 Results and Analysis

In this competition, each team has up to five submission opportunities on the final test set. We re-

[9] https://github.com/alexa/dstc11-track5/tree/main/data

[10] https://github.com/alexa/dstc11-track5

[11] https://github.com/huggingface/transformers

Models	On validation set				On final-test set			
	Precision	Recall	F1	EM	Precision	Recall	F1	EM
DeBERTa-v3-base (baseline)	0.7951	0.8843	0.8373	0.4049	0.7901	0.7877	0.7889	0.3906
DeBERTa-v3-base (seen expert)	0.7966	0.8841	0.8379	0.4224	-	-	-	-
RoBERTa-large (unseen expert)	0.7896	0.8410	0.8252	0.4110	-	-	-	-
DeBERTa-v3-large (de-noise expert)	0.7996	0.8713	0.8358	0.4310	-	-	-	-
Ensemble with Precision	**0.8894**	0.8434	0.8658	0.4937	-	-	-	-
Ensemble with Recall	0.7278	**0.9486**	0.8236	0.4227	-	-	-	-
Ensemble with F1	0.8591	0.8889	**0.8737**	0.5148	0.8096	0.8413	0.8252	0.5210
Ensemble with EM	0.8588	0.8763	0.8675	**0.5195**	**0.8183**	**0.8506**	**0.8342**	**0.5314**

Table 4: Experimental results on the knowledge entry selection. EM is short for exact match.

port the submission results and also provide the performance on the validation set.

5.1 Sub-task 1: knowledge seeking turn detection

Table 2 shows the experimental results. We can see the baseline model already has a high performance on the validation set. However, considering there are unseen instances in the final test set, we aim to use the difference-aware method to have a better performance. We test the ensemble with Precision, Recall, and F1 and choose the last two as our final submission since a higher Recall is more useful for unseen instances. On the final test set, our method achieve better Recall and F1 than the baseline model (the results of the baselines are reported by the organizer). Specifically, we have the highest F1 and second-highest Recall among all submissions. We also have the highest sum of the Precision/Recall/F1. These results give us an advantage over the rest of the sub-tasks.

5.2 Sub-task 3: knowledge entry selection

Table 4 shows the experimental results of the knowledge entry selection task. We can see that the experts we trained all performed close to the baseline. They are only a little better on the exact match metric than the baseline. However, when using our ensemble method, the performance is largely improved. On the validation set, the highest Precision, Recall, F1, and EM results are obtained when using the corresponding metric as the ensemble indicator. These results show that 1) the different expert models are good at different aspects; 2) our ensemble method can successfully combine the advantage of the experts and achieve the desired results on specific metrics. We finally choose the last two results (ensemble with F1 and EM) for final submission according to the sum of all four metrics. On the test set, our method has consistent performance and largely outperforms the baseline. On EM, our method surpasses the baseline by 14 percent. We denote these two results as KS-F1 and KS-EM for the convenience of the next section.

5.3 Sub-task 4: response generation

Table 5 shows the experimental results on response generation. We only have five submission opportunities and all the results are shown in the Table. Notice that we did not use any ensemble in a generation. The results can reflect how the knowledge selection results affect the generation. The baseline uses the selected knowledge from the baseline in Table 4. The BART-base with KS-F1 use the fourth result of Table 4), its generation results are better than baselines on all metrics. This result shows that the KS-F1 provides higher-quality knowledge entries and again proves the effectiveness of our ensemble methods. Benefiting from more parameters, the BART-large and T5 are better than BART-base on most metrics. The BLEU of BART-large (KS-F1) ranks 2nd among all the submissions to track-5. The BART-large (KS-EM) is better on ROUGE and the T5-3B (KS-EM) is better on METEOR. However, the T5 models perform badly on BLEU on the test set and do not show an obvious advantage over BART-large. This may indicate that fine-tuning pre-trained language models with large amounts of parameters to a specific task is not always work. The DSTC 11 track-5 also performs a manual evaluation for the generation results. The organizer only chooses one of the submissions for each team based on the mean reciprocal rank over all the metrics. The BART-large (KS-F1) is chosen by the organizer since this submission has the highest overall score among our 5 submissions. We finally ranked 5th among all teams in this manual evaluation. In our own manual evaluation, the BART-large (KS-EM) is better than the BART-large (KS-F1). We also provide the results on LLAMA-13B, the results are much lower than other models.

Models	On validation set			On final-test set		
	BLEU	METEOR	ROUGE	BLEU	METEOR	ROUGE
BART-base (baseline)	0.1042	0.1810	0.3651 / 0.1506 / 0.2875	0.1004	0.1748	0.3520 / 0.1430 / 0.2753
LLAMA (fine-tuned)	0.0653	0.1012	0.2132 / 0.0975 / 0.1667	-	-	-
BART-base (KS-F1)	0.1089	0.1793	0.3689 / 0.1532 / 0.2918	0.1047	0.1764	0.3603 / 0.1463 / 0.2801
BART-large (KS-F1)	0.1087	0.1796	0.3693 / 0.1530 / 0.2925	**0.1075**	0.1744	0.3585 / 0.1459 / 0.2794
BART-large (KS-EM)	0.1103	0.1796	0.3695 **/ 0.1534 / 0.2929**	0.1050	0.1774	**0.3617 / 0.1474** / 0.2805
T5-3B (KS-F1)	**0.1104**	0.1773	0.3657 / 0.1524 / 0.2883	0.0897	0.1744	0.3591 / 0.1458 / **0.2840**
T5-3B (KS-EM)	0.1042	**0.1827**	**0.3743** / 0.1489 / 0.2894	0.0959	**0.1805**	0.3552 / 0.1450 / 0.2784

Table 5: Experimental results on the response generation.

6 Conclusion

We participated in track-5 of the 11th Dialog System Technology Challenges (DSTC 11), called Task-oriented Conversational Modeling with Subjective-knowledge. The task includes four subtasks and we propose a difference-aware ensemble method to address two main challenges in this competition. The first challenge is there are multiple reasonable knowledge entries for one context. The second challenge is that the data distribution is different in train/validation/test sets. We first train several expert models that are good at certain aspects for each sub-task. Then we use the difference-aware ensemble method to balance the abilities of expert models. Experimental results verify the effectiveness of our method and we got third on the final test set. Future work includes but is not limited to 1) testing our ensemble method on the knowledge entity tracking task; 2) designing a method to automatically learned the threshold in Algorithm 1.

Limitations

The method proposed by this work is only verified on limited data in DSTC competition and the language is only with narrow morphology (English). Whether the technique can be used for other morphology needs further verification. Another limitation is that the SK-TOD generation task needs to explicitly consider the diversity of opinions and the proportion of sentiments. This requires a specific module to assist the generation model or using a much larger language model such as GPT-4 to fully utilize the review information. We also do not perform these experiments and hope future researchers could investigate this task.

Ethics Statement

The dataset we used is from the DSTC 11 track-5. We use the back-translation and synonym substitution to construct a noise version of the data for training. The generation models trained with this noise data may learn to generate semantically incorrect or unfriendly responses.

Acknowledgements

This paper is supported by the Science and Technology Innovation 2030 Major Project of China (No. 2021ZD0113302) and National Natural Science Foundation of China (No. 62076081, No. 61772153 and No. 61936010).

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. In *ACL (1)*, pages 1308–1319. Association for Computational Linguistics.

Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3365–3373.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Mudit Chaudhary, Borislav Dzodzo, Sida Huang, Chun Hei Lo, Mingzhi Lyu, Lun Yiu Nie, Jinbo Xing, Tianhua Zhang, Xiaoying Zhang, Jingyan Zhou, Hong Cheng, Wai Lam, and Helen Meng. 2021. Unstructured knowledge access in task-oriented dialog modeling using language inference, knowledge retrieval and knowledge-integrative response generation. *CoRR*, abs/2101.06066.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul A. Crook, and William Yang Wang. 2022. KETOD: knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA,* *United States, July 10-15, 2022*, pages 2581–2593. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Song Feng, Hui Wan, R. Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis A. Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8118–8128. Association for Computational Linguistics.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. pages 1891–1895. ISCA.

Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Yoonna Jang, Jungwoo Lim, Yuna Hur, Dongsuk Oh, Suhyune Son, Yeonsoo Lee, Dong-Hoon Shin, Seungryong Kim, and Heuiseok Lim. 2022. Call for customized conversation: Customized conversation grounding persona and knowledge. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10803–10812. AAAI Press.

Di Jin, Seokhwan Kim, and Dilek Hakkani-Tur. 2021. Can I be of further assistance? using unstructured knowledge access to improve task-oriented conversational modeling. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering, DialDoc@ACL-IJCNLP 2021, Online, August 5, 2021*, pages 119–127. Association for Computational Linguistics.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020a. Sequential latent knowledge selection for knowledge-grounded dialogue. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tür. 2020b. Beyond domain apis: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 278–289. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *WMT@ACL*, pages 228–231. Association for Computational Linguistics.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Trans. Speech Audio Process.*, 8(1):11–23.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian J. McAuley. 2022. Achieving conversational goals with unsupervised post-hoc knowledge injection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3140–3153. Association for Computational Linguistics.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2775–2779. Association for Computational Linguistics.

Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and Maarten de Rijke. 2020. Dukenet: A dual knowledge interaction network for knowledge-grounded conversation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1151–1160. ACM.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *EMNLP*, pages 2322–2332. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Shrimai Prabhumoye, Kazuma Hashimoto, Yingbo Zhou, Alan W. Black, and Ruslan Salakhutdinov. 2021. Focused attention improves document-grounded generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4274–4287. Association for Computational Linguistics.

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

Chao-Hong Tan, Xiaoyu Yang, Zi'ou Zheng, Tianda Li, Yufei Feng, Jia-Chen Gu, Quan Liu, Dan Liu, Zhen-Hua Ling, and Xiaodan Zhu. 2020. Learning to retrieve entity-aware knowledge and generate responses with copy mechanism for task-oriented dialogue systems. *CoRR*, abs/2012.11937.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Lin Xu, Qixian Zhou, Jinlan Fu, Min-Yen Kan, and See-Kiong Ng. 2022. Corefdiffs: Co-referential and differential knowledge flow in document grounded conversations. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 471–484. International Committee on Computational Linguistics.

Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *AAAI*, pages 4618–4626. AAAI Press.

Chao Zhao, Spandana Gella, Seokhwan Kim, Di Jin, Devamanyu Hazarika, Alexandros Papangelis, Behnam Hedayatnia, Mahdi Namazifar, Yang Liu, and Dilek Hakkani-Tur. 2023. "what do others think?": Task-oriented conversational modeling with subjective knowledge. *CoRR*, abs/2305.12091.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskénazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. In *SIGDIAL Conference*, pages 27–36. Association for Computational Linguistics.

Xueliang Zhao, Tingchen Fu, Chongyang Tao, and Rui Yan. 2022. There is no standard answer: Knowledge-grounded dialogue generation with adversarial activated multi-reference learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1878–1891. Association for Computational Linguistics.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3377–3390. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. 2018. A dataset for document grounded conversations. In *EMNLP*, pages 708–713. Association for Computational Linguistics.