Ensemble Method via Ranking Model for Conversational Modeling with Subjective Knowledge

Xin Huang, Kye Min Tan, Richeng Duan, Bowei Zou
Institute for Infocomm Research (I2R), A*STAR, Singapore
{huangx2,tan_kye_min,Duan_Richeng,zou_bowei}@i2r.a-star.edu.sg

Abstract

This paper describes our submission to the fifth track of the 11th Dialog System Technology Challenge (DSTC11), which focuses on "Task-oriented Conversational Modeling with Subjective Knowledge". We focus on response generation and leverage a ranking strategy to ensemble individual models of BART, Long-T5, and a fine-tuned large language model based on LLaMA. The strategy is supplemented by other techniques like low rank adaptation to maintain efficient utilization of these large models while still achieving optimal performance. The experiments show that the ensemble method outperforms individual models and the baseline method. Our model was ranked 1st place in ROUGE_1, 2nd place in ROUGE_L score and 4th place in human evaluation among a total of 14 participating teams.

1 Introduction

The task of developing effective and engaging task-oriented dialogue systems has been a subject of ongoing research and development in the field of natural language processing (NLP). Traditional approaches to task-oriented dialogue systems have focused on providing information and performing actions based on predefined rules or accessing back-end databases or APIs. However, user requests may sometimes require information outside the scope of these structured resources, requiring the system to incorporate additional domain knowledge from external unstructured sources.

Past challenges such as DSTC9 track 1 (Kim et al., 2021a) and DSTC10 track 2 (Kim et al., 2021b) have proposed the use of unstructured knowledge from FAQs to build a knowledge-grounded task-oriented dialogue system. The tasks proposed in DSTC11 track 5 build off past work by including subjective knowledge from customer reviews, which also involves addressing the complexity of differentiating multiple aspects within a single review post, conflicting opinions, and comparison requests between entities.

We primarily focus on the third sub-task of this challenge, which involves response generation. Our approach combines the utilization of large foundational language models with a ranking strategy for response prioritization. Additionally, we employ the method of low rank adaptation (LoRA) (Hu et al., 2021) to keep resource usage low while training and using such large models. To enhance generation performance, we employ an ensemble technique that combines multiple individual models, leveraging the capabilities of GPT-4 (OpenAI, 2023) in the process. Our final model achieved impressive rankings in the evaluation metrics. It secured the 1st place in ROUGE_1, the 2nd place in ROUGE_L score, and the 4th place in human evaluation. This demonstrates the effectiveness and competitiveness of our approach in the challenge.

2 Related Work

Several methods of improving knowledge-grounded task-oriented dialogue systems have been explored by participants in past DSTC challenges. Thulke et al. (2023) implemented a noisy channel model to separate the tasks of generating a response and incorporating knowledge by reranking multiple candidate responses. He et al. (2021a) and Tian et al. (2021) both use large pre-trained language models exceeding 1 billion parameters to generate the final response. The former focuses on incorporating structured knowledge from MultiWOZ 2.2 (Zang et al., 2020) and negative sampling for the knowledge selection task, while the latter explores several approaches to generating more synthetic dialogue data.

The scaling up of large language models in recent years has continued to show gains in terms of performance and abilities (Wei et al., 2022). However, the prohibitive cost of fine-tuning all the parameters in these models has led to the exploration

Proceedings of The Eleventh Dialog System Technology Challenge, pages 177–184
September 11, 2023 ©2023 Association for Computational Linguistics

of parameter efficient fine-tuning strategies, such as adapter layers (Houlsby et al., 2019), prompt tuning (Liu et al., 2021; Li and Liang, 2021; Liu et al., 2022), and LoRA (Hu et al., 2021).

3 Task Description

The aim of this challenge track is to generate a relevant response given a set of user and agent turns, and a knowledge base which may contain information relevant to the user's query. This is further divided into 3 sub-tasks:

1. Knowledge Turn Detection: Determining whether a given query is knowledge-seeking and should be handled by the following steps
2. Knowledge Selection: Selecting knowledge snippets consisting of user reviews and FAQs relevant to the query
3. Response Generation: Generating a response to the query based on knowledge snippets selected in the previous step

The training data provided for this challenge includes 28,431 dialogues, of which 14,768 are knowledge-seeking turns which will be processed by the knowledge selection and response generation steps. A knowledge base containing subjective knowledge in the form of user reviews and objective knowledge in the form of FAQs is also provided. The dialogues are augmented from MultiWoz 2.1 to include additional knowledge-seeking turns which require information from the knowledge base. All dialogues and knowledge snippets belong to either the hotel or restaurant domain. Table 1 shows the quantities of dialogue data and knowledge for these two domains. Additionally, each knowledge snippet is labeled with the name of the entity (hotel or restaurant) it refers to, the reviewer profile for user reviews (solo, couple, family, friends, colleagues), and the food and drink items mentioned in restaurant reviews. On average, each knowledge seeking turn in the train set has 3.80 relevant knowledge snippets, while the validation set has 4.07 relevant knowledge snippets.

4 Methodology

Figure 1 depicts the overall architecture of our methods.

Type	Hotel	Rest.
Dialogues (Train)	7859	6909
Dialogues (Valid)	1436	693
Entities	33	110
Reviews	330	1100
FAQs	1219	1650

Table 1: Quantity of data by domain.

4.1 Knowledge Seeking Detection (KTD)

In sub-task 1, which focuses on knowledge-seeking turn detection, we employ the baseline method which utilizes a DeBERTa (He et al., 2021b) model to encode the concatenated input of the dialogue context $C = [U_1, S_1 ... U_t, S_t]$, including user utterance U_i and system response S_i at each step of the conversation. To classify whether the current user utterance contains a knowledge-seeking request, we utilize a binary classifier, which leverages the encoded representation of the dialogue context produced by the DeBERTa model to make predictions.

$$h = DeBERTa(C)$$

$$P(C) = softmax(Dense(h))$$

4.2 Knowledge Entity Matching

The goal of knowledge entity matching process is to select entities $E = \{e_1, ..., e_m\}$ which are relevant to the user utterance given dialogue context C and a list of knowledge snippets. We follow the baseline which uses a word-matching method based on Jin et al. (2021) for entity extraction. Specifically, the method first applies a list of heuristic rules which are used to normalize entity names, then uses n-gram fuzzy matching between the normalized entity names and all dialogue turn utterances. The longest contiguous matching subsequence (LCS) algorithm is used to calculate the similarity between a entity name and a user utterance, and a threshold of 0.95 is used to determine whether an entity is considered a match. Finally, the method selects the entities from the last dialogue turn in which entities are detected.

4.3 Knowledge Selection

The knowledge selection task aims to select knowledge snippets which are relevant to the current user utterance at each turn given dialogue context C and a list of candidate knowledge snippets $K^E = \{k_1^E, ..., k_n^E\}$ where each element of the

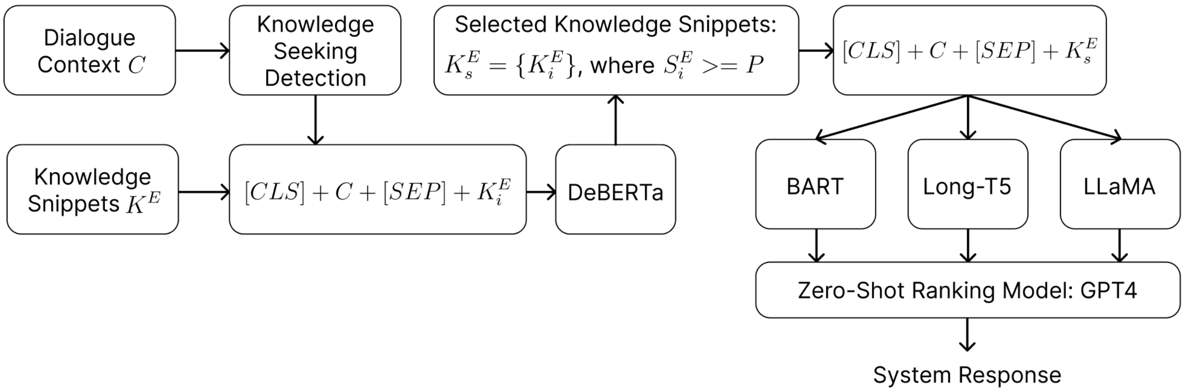

Figure 1: The overall architecture of our methods

knowledge snippets belongs to the selected entities E during the entity matching process.

We consider the knowledge selection process as a sequence classification task. Formally, for each turn, the method first concatenates the dialogue history with each candidate knowledge snippet k_i^E from the matched knowledge entities E. A DeBERTa encoder is used to encode the concatenated input to obtain hidden representation h_i and then a classification head followed by a softmax function is used to obtain the relevance probability of the candidate knowledge snippet.

$$U_i^E = [C, K_i^E]$$
$$h_i^E = Deberta(U_i^E)$$
$$S_i^E = Softmax(Dense(h_i^E))$$

During training, a sampling approach is used to reduce the number of negative candidate snippets from the large candidate space in the knowledge base. Following the baseline, we only sample negative candidates from entities which are relevant to the dialogue. A binary cross-entropy loss is used to optimize the model. As the number of knowledge snippets required for each dialogue varies, a threshold P is applied to the relevance probabilities to select candidate snippets during inference. P is estimated based on the validation set. As the optimal P for the test set may be different, we dynamically vary the threshold during inference by lowering it if no knowledge candidates meet the original threshold. We repeat this process until at least one knowledge snippet is selected.

4.4 Response Generation

The response generation task aims to generate a proper system response for user requests for knowledge given dialogue context C and selected knowledge snippets K_s^E. We first adapt the following transformer based models including encoder-decoder models and decoder-only models to generate a system response based on the concatenated input of dialogue context C and concatenated knowledge snippets K_s^E as a single string:

1. BART model (Lewis et al., 2020): the same model used in the baseline, we increase the maximum input knowledge token size from 256 to 512 to avoid knowledge cutoff for some turns.

2. Long-T5 model (Guo et al., 2022): Long-T5 model purposes local attention and transient global attention methods which show improved performance for long sentence generation tasks such as summarization. We hypothesize that a model with strong summarization capabilities will be beneficial for combining information across multiple user reviews, thus improving the quality of the response.

3. LLaMA model (Touvron et al., 2023): The LLaMA model is a foundation decoder-only model which is pre-trained on more than 1 trillion tokens. Recent research like Peng et al. (2023) also show that the LLaMA model is capable of GPT-4 like performance after instruction tuning. In this challenge, we explore directly fine-tuning the original LLaMA model.

We first train all 3 models including BART, Long-T5 and LLaMA model individually with all 3 models using the same input from the knowledge-seeking detection and knowledge selection task. For BART and T5 model, we reuse the baseline approach to train the model while only changing the tokenizer and encoder-decoder model. For LLaMA model, we use the parameter efficient tuning method LoRA for model fine-tuning which can avoid memory issues using limited GPU resources. We train the model to maximize next token probability via aligned language modeling.

4.5 Response ranking

To encourage diversity of the system response and utilize strong points from different models, we propose an ensemble method for generating the final response via ranking of the three models by a ranking model.

The goal of the ranking model is to give scores S_j for the quality of the response given the concatenated input of dialogue context C, concatenated knowledge snippets K_s^E and output of each generation model $o_j, j \in \{1, 2, 3\}$. We designed a customized prompt p_c for the independent model in order to perform zero-shot evaluation on the response generation by different models. We make use of a large language model GPT-4 and use their API for the evaluation.

$$S_j = GPT4([p_c, C, K_s^E, o_j]), j \in \{1, 2, 3\}$$

Then we ensemble the model outputs using two methods: 1) select the model output with the highest score for each turn. 2) select the model output with the highest score only if the output of the best reference model scores below or equal to a threshold S_t, otherwise we select the output of the best reference model instead. We consider the best reference model using human evaluation on the dev-set with a likert scale of 1 to 5 by sampling N utterances as our best reference model.

5 Experiments

5.1 Sub-task 1: Knowledge-Seeking Turn Detection (KTD)

For the knowledge turn detection task, we use the baseline implementation of a DeBERTa v3 model with a binary classifier.

Data: For training, we use all 28431 dialogue samples in the provided dataset. For validation, we use all 4173 dialogue samples in the dataset.

Metrics: The evaluation metrics for KTD are precision, recall, and F1.

Hyperparameters: We use the baseline hyperparameters of learning rate $\alpha = 3 * 10^{-5}$, number of epochs $E = 10$, maximum history token size of 510, and Adam optimizer with $\epsilon = 10^{-8}$ and $\beta_1 = 0.9, \beta_2 = 0.999$.

5.2 Sub-task 2: Knowledge Selection (KS)

For the knowledge selection task, we also use the DeBERTa model as a cross encoder followed by a classifier as implemented in the baseline. During inference, we begin by filtering knowledge snippets by the probability estimated by the model using the threshold P, and repeatedly lower it by 0.5 if none of the candidates meet the threshold.

Data: For training, we use 14768 dialogue samples which require knowledge access. The validation set consists of 2129 dialogue samples.

Metrics: The evaluation metrics for KS are precision, recall, and F1 on the snippet level, where the metrics are calculated across all <C, K_S> pairs.

Hyperparameters: We use the baseline hyperparameters of learning rate $\alpha = 3 * 10^{-5}$, number of epochs $E = 3$, maximum history and knowledge token size of 256 each, and Adam optimizer with $\epsilon = 10^{-8}$ and $\beta_1 = 0.9, \beta_2 = 0.999$.

5.3 Sub-task 3: Response Generation (RG)

For RG, our initial experimentation involves generating responses independently from individual models, including BART, Long-T5, and LLaMA. To further enhance the quality and effectiveness of the responses, we employ an ensemble approach. We utilize GPT-4 to score the responses generated by the individual models for each dialogue turn. The scores assigned by GPT-4 are used to determine the most suitable response from the ensemble of models, which allows us to leverage the strengths of each individual model while mitigating their weaknesses.

Data: Similar to the KS task, we use 14768 training samples and 2129 validation samples which have knowledge and system response labels.

Metrics: The automatic evaluation metrics for RG task include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). Apart from the official human evaluation, we also internally assess the results of different models for aspect accuracy.

Hyperparameters: For the Bart and Long-T5 model, we chose the same hyperparameters of learning rate $\alpha = 5 * 10^{-5}$, number of epochs $E = 3$, maximum history and knowledge token size of 512, and we use the Adam optimizer with $\epsilon = 10^{-8}$ and $\beta_1 = 0.9, \beta_2 = 0.999$. We select transient-global attention for Long-T5 model.

For the LLaMA model we select the pretrained LLaMA model with 7 billion parameters, and we chose LoRA rank $r = 8$ which applies to W_q, W_v for this experiment. For the ensembled model, we set the reference model as LLaMA model because it scores highest based on our human evaluation on the dev-set. We chose ensembling score threshold $S_t = 3$ with a scoring likert scale of 1 to 5.

6 Results and Analysis

We present our experimental results and analyses in this section.

6.1 Sub-task 1: Knowledge-Seeking Turn Detection (KTD)

Data Split	**P**	**R**	**F1**
Validation	99.92	99.95	99.93
Test	99.86	99.79	99.82

Table 2: Evaluation results for Task 1 on validation and test set.

Data		**Metrics**		
Train	Val	P	R	F1
H	H	99.86	99.86	99.86
	R	99.33	86.00	92.19
R	H	99.91	79.46	88.52
	R	99.86	100.0	99.93
All	H	100.0	99.86	99.93
	R	99.86	99.86	99.86

Table 3: Analysis of domain masking for Task 1 on the validation set. H and R represent hotel and restaurant domains respectively

The performance of the baseline model for sub-task 1 is presented in Table 2. Although the model demonstrates near-perfect performance on both the validation and test sets, we conducted an examination of its generalization capability through domain masking in this work. Table 3 shows the performance of the model in the cross-domain evaluation setting. The results reveal a substantial decline in performance when validating on the masked domains, especially in terms of recall, suggesting that the model's effectiveness diminishes when confronted with unseen domains. This drop in performance underscores the challenge of domain adaptation in KTD, as in most NLP tasks. These findings also emphasize the need to carefully consider domain adaptation during test set construction and further explore this aspect in future research endeavors.

6.2 Sub-task 2: Knowledge Selection (KS)

Model	**P**	**R**	**F1**	**EM**
Fixed P	**77.11**	82.01	79.48	42.39
Dynamic P	77.03	**82.17**	**79.52**	**42.47**

Table 4: Evaluation results for Task 2 on test set.

Data		**Metrics**			
Train	Val	P	R	F1	EM
H	H	72.94	92.99	81.75	32.34
	R	33.59	83.79	47.96	18.59
R	H	72.21	80.56	76.15	28.44
	R	73.22	78.90	75.96	35.73
All	H	81.64	94.39	99.93	43.32
	R	68.59	80.70	74.15	36.46

Table 5: Analysis of domain masking for Task 2 on the validation set. H and R represent hotel and restaurant domains respectively

We analyze the effect of dynamically adjusting the threshold, P for KS. Using a fixed P obtained from tuning on the validation set, we find that no knowledge snippets are retrieved for 31 out of 2796 samples filtered by the KTD step. We observe a small uplift in recall, F1, and exact match metrics in the test set after using a dynamic P value, as shown in Table 4.

As with the KTD task, we also evaluated the KS model by domain masking, presented in Table 5. We observe that masking domains in this case also negatively impacts the performance of unmasked domains, which may be due to the limited amount of training data used. We also note that performance on the restaurant domain is comparatively weaker, which may be due to the larger number of knowledge snippets (as shown in Table 1) and the additional aspect of food types compared to the hotel domain. Hence, further improvement to the KS task may be obtained by focusing on improving the quality or availability of restaurant domain data.

6.3 Sub-task 3: Response Generation (RG)

Table 7 provides a summary of the automatic evaluation results for various models on the task 3 test set. Notably, each individual model exhibits its own strengths: Long T5 achieves the highest BLEU

U: Is there any restaurants located in the north side of town? S: There is one. It is da vinci pizzeria. Shall I book it? U: Does Da Vinci Pizzeria serve portions that are plentiful?
BART: Yes, Da Vinci Pizzeria does serve portions that are plentiful. Past guests have said that the portions are plentiful. Would you like to know more about them? **Long-T5**: The Da Vinci Pizzeria serves plentiful portions, but some customers have said that the portions are too small. **LLaMA**: Yes, Da Vinci Pizzeria does serve portion sizes that are plentiful. Would you like to know more about them? **E2.**: The Da Vinci Pizzeria serves plentiful portions, but some customers have said that the portions are too small. **GPT-4 (standalone)**: Some customers feel that Da Vinci Pizzeria serves portions that are plentiful and satisfying, while others think the portions are relatively small. It can vary depending on personal preferences and appetites. If you enjoy hearty portions, you might want to explore other options or consider ordering extra items to ensure you're fully satisfied.
Reference: Some past customers do feel that the portions are plentiful, but there are others who felt that they are too small.

Table 6: Sampled output of our different models. For GPT-4 (standalone), we directly instruct prompting the GPT-4 using input of dialogue context C and selected knowledge K_s^E for answer generation.

Model	BLEU	MT	R-1	R-2	R-L
Baseline	10.04	17.48	35.20	14.30	27.53
Long T5	**10.50**	16.91	34.72	14.66	27.93
BART	10.20	17.50	35.28	14.34	27.60
LLaMA	9.44	17.47	36.52	14.94	**28.75**
E1.	9.84	17.74	**36.58**	**15.09**	**28.75**
E2. ($S_t = 3$)	10.24	**17.76**	36.33	15.08	28.65

Table 7: Automatic evaluation results on test-set for Task 3. Evaluation metrics including BLEU-4 (BLEU), ROUGE (R-1, R-2, R-L), METEOR (MT). E1. stands for our first ensemble method, and E2. stands for the second ensemble method with threshold $S_t = 3$.

Model	**Score**		**PCC**
	Human	**GPT-4**	
BART	3.64	4.12	0.3626
Long T5	3.36	4.07	0.2856
LLaMA	3.88	4.28	0.2884
E1.	**3.90**	4.52	0.1785
E2. ($S_t = 3$)	3.82	**4.73**	0.3026

Table 8: Comparison between the average scores of our internal human evaluation and GPT4 evaluation for Task 3, and Pearson correlation coefficients (PCC).

score, BART excels in METEOR, and LLaMA performs best in terms of ROUGE score. Furthermore, we observe that the performance of $E1.$ closely approaches that of the reference model LLaMA, suggesting that a naive ranking method may not outperform the strongest individual model. The method $E2.$ with $S_t = 3$ yields the best overall performance and we use Table 6 to showcase sampled outputs from different models. It is evident that the results generated by BART and LLaMA lack some negative opinions, while Long T5 effectively summarizes the reviews in this case. The ensemble model successfully selects Long T5 as the final output, thereby maintaining the appropriateness of the response compared to the standalone zero-shot response generation by GPT-4.

Table 8 illustrates the results obtained from both human assessors and GPT-4 for our models on the test set for RG. For human evaluation, we enlist the expertise of four NLP experts who employ a Likert scale, ranging from 1 to 5, to rate the responses. The experts are provided with the dialogue context, oracle knowledge snippets, and predicted response for each model. To ensure consistency and reliability, we randomly sample a total of 120 instances and divide them into four groups. Among these groups, each pair has a 30% overlap, enabling us to assess the consistency of scores. The scores assigned by the experts aim to evaluate the aspect accuracy of the response. To determine the level of agreement among the experts, we calculate the average score difference. This amounts to 0.3, which indicates a mutual agreement regarding the scoring of the response, ensuring a reliable evaluation process.

From Table 8, we first see the two ensemble methods outperform individual models in terms of both human and GPT-4 ratings, suggesting that the responses generated by the ensemble approach are better. We can also observe that GPT-4's scores generally align with the scores given by the human evaluators for the different models, which indicates that GPT-4 possesses a reasonable ability to assess Task 3, though not completely consistent with human judgments. We also calculate Pearson correlation coefficients using the equation:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

, where n is the number of samples in the test set, X is human evaluators' score and Y is GPT-

4's score, for each of the models and ensemble methods, which shows that GPT-4's and the human evaluators' judgement is weakly correlated. Further investigation of individual samples with negative correlation reveals cases where both the human raters and GPT-4 assigned scores inconsistent with judging criteria. For future evaluations, well-defined judging criteria and experimenting with different prompts may improve the accuracy and consistency of scores.

GPT-4's scoring is based on its training data and may excel in assessing certain aspects such as grammar or syntax but might struggle with understanding context-dependent nuances that human evaluators are typically adept at capturing. The alignment between the scores assigned by GPT-4, human evaluations, and the ROUGE scores obtained through automatic evaluation in Table 7 highlights an intriguing consistency. This suggests that ROUGE can offer valuable insights into the correctness of text generation. Nevertheless, for a comprehensive assessment, it is crucial to consider multiple evaluation criteria and perspectives when evaluating response generation models.

Model	Approp.	Asp-Acc	Average
Baseline	3.6348	2.8715	3.2531
Best Team	**3.6596**	**2.9095**	**3.2846**
E2. ($S_t = 3$)	3.6487	2.8908	3.2697

Table 9: Official human evaluation results for Task 3.

The official human evaluation results for this challenge are presented in Table 9, assessing the appropriateness of the response and aspect accuracy as evaluation metrics. Our submission of the ensemble method with $S_t = 3$ achieved the 4th place in the official human evaluation. However, the top-performing model only exhibited a marginal improvement of 0.0149 in terms of average score compared to our method. We attribute our performance gap relative to the best model primarily to our weaker performance in task 2, where the best model demonstrated a significant 4.2% absolute improvement over our submission in terms of F1 score for knowledge snippet selection. The incorrect selection of knowledge snippets can adversely impact the performance of both individual models and the ranking model.

7 Conclusion

In this work, we demonstrated an ensemble method that utilizes a ranking strategy to combine outputs from various large models for response generation. Our submission to the DSTC11 Track 5 challenge achieved the highest Rouge_1 score and secured the 2nd place for Rouge_L score in sub-task 3, despite relatively weaker performance in sub-task 1 and 2. Furthermore, we conducted an analysis on domain masking for sub-task 1 and 2 to assess performance in handling requests from unseen domains. Our findings highlight the need for further exploration to improve performance when the system encounters domains that significantly differ from the training data.

8 Limitations

Currently, our method uses GPT-4 for ranking individual models, but this introduces a dependency on an external service that may not always be available. In our future work, we intend to explore the utilization of an easily accessible ranking model to mitigate this dependency.

Additionally, our method requires a list of different models for result ensembling which further adds to inference cost. We plan to investigate the usage of a single open-source large language model for both answer generation and self evaluation in future work.

As with most large language models currently published, we are unable to guarantee the safety and freedom from bias of its output as it was trained on clean data, and further work in ensuring safety is recommended before the system is used as part of a large scale operation.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

H. He, Hua Lu, Siqi Bao, Fan Wang, Hua Wu, Zhengyu Niu, and Haifeng Wang. 2021a. Learning to select external knowledge with multi-scale negative sampling. *ArXiv*, abs/2102.02096.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

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 2021)*, pages 119–127, Online. Association for Computational Linguistics.

Seokhwan Kim, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, and Dilek Hakkani-Tur. 2021a. Beyond domain apis: Task-oriented conversational modeling with unstructured knowledge access track in dstc9.

Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papangelis, Karthik Gopalakrishnan, Behnam Hedayatnia, and Dilek Hakkani-Tur. 2021b. "how robust r u?": Evaluating task-oriented dialogue systems on spoken conversations.

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*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4.

David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2023. Task-oriented document-grounded dialog systems by hltpr@rwth for dstc9 and dstc10.

Xin Tian, Xinxian Huang, Dongfeng He, Yingzhan Lin, Siqi Bao, Huang He, Liankai Huang, Qiang Ju, Xiyuan Zhang, Jian Xie, Shuqi Sun, Fan Wang, Hua Wu, and Haifeng Wang. 2021. Tod-da: Towards boosting the robustness of task-oriented dialogue modeling on spoken conversations.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.