

# Adapting Text-based Dialogue State Tracker for Spoken Dialogues

Jaeseok Yoon<sup>1\*</sup>, Seunghyun Hwang<sup>1\*</sup>, Ran Han<sup>2</sup>, Jeonguk Bang<sup>2</sup>, Kee-Eung Kim<sup>1,3</sup>

<sup>1</sup>Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea

<sup>2</sup>Electronics Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

<sup>3</sup>School of Computing, KAIST, Daejeon, Republic of Korea

{jake.yoon, steven1971}@kaist.ac.kr, {ran.han, jubang0219}@etri.re.kr, kekim@kaist.ac.kr

## Abstract

Although there have been remarkable advances in dialogue systems through the dialogue systems technology challenge (DSTC), it remains one of the key challenges to building a robust task-oriented dialogue system with a speech interface. Most of the progress has been made for text-based dialogue systems since there are abundant datasets with written corpora while those with spoken dialogues are very scarce. However, as can be seen from voice assistant systems such as Siri and Alexa, it is of practical importance to transfer the success to spoken dialogues. In this paper, we describe our engineering effort in building a highly successful model that participated in the speech-aware dialogue systems technology challenge track in DSTC11. Our model consists of three major modules: (1) automatic speech recognition error correction to bridge the gap between the spoken and the text utterances, (2) text-based dialogue system (D3ST) for estimating the slots and values using slot descriptions, and (3) post-processing for recovering the error of the estimated slot value. Our experiments show that it is important to use an explicit automatic speech recognition error correction module, post-processing, and data augmentation to adapt a text-based dialogue state tracker for spoken dialogue corpora.

## 1 Introduction

Task-oriented dialogue (TOD) systems aim to successfully perform various tasks, such as hotel reservations, train information retrieval, and food ordering, by interacting with users. In particular, with the popularity of voice assistant systems such as Siri and Alexa, it is important to be able to take spoken languages as inputs to TOD systems. These TOD systems with the speech interface is referred to as *spoken dialogue systems*.

However, due to the differences in many aspects between text conversation and speech conversation,

the existing TOD systems for text conversation can fail to achieve good performance for speech conversation. To this end, several tasks such as ATIS (Hemphill et al., 1990) and competitions (DSTC2<sup>1</sup>, DSTC10 knowledge-grounded task-oriented dialogue modeling on spoken conversations track<sup>2</sup>) were proposed. In the knowledge-grounded task-oriented dialogue modeling on spoken conversations track in DSTC10, various methods were applied to spoken dialogue systems. Khan et al. (2015) introduced a method to reduce automatic speech recognition (ASR) error by making predictions using multiple hypotheses of several ASR models together. Also, Tian et al. (2021) proposed a method to reduce the discrepancy between written conversation and spoken conversation by augmenting the dataset based on ontology and pattern or generating noisy addition of spoken conversation through a text-to-speech (TTS)->ASR pipeline.

Since only ASR hypothesis data was provided in knowledge-grounded task-oriented dialogue modeling on spoken conversations track in DSTC10, it was difficult to effectively deal with errors generated in ASR, and most of the studies were focused on data augmentation to make up for the lack of data. On the other hand, in the speech-aware dialogue system technology track in DSTC11, in addition to the ASR hypothesis data, audio, transcripts used to generate the audio and augmented data were provided, allowing a more diverse approach. For this motivation, we studied how to solve problems caused by the propagation of ASR errors to the model and how to solve errors such as incorrect proper nouns generated by spoken input and participated in the speech-aware dialogue systems technology challenge track in DSTC11.

In this paper, we propose a model, which allows good performance for spoken utterance input as

<sup>1</sup><https://github.com/matthen/dstc>

<sup>2</sup><https://github.com/alexa/alexa-with-dstc10-track2-dataset>

\*These authors contributed equally.

Speaker	Utterance	Dialog state
User	I'd like to find out if there are any 4-star rated <b>guesthouses</b> in cambridge	hotel-stars=4 hotel-type=guesthouse hotel-internet=yes restaurant-name=bangkok city restaurant-area=centre
System	There are quite a few. what area do you prefer?	
User	I'm open to any area as long as there is <b>free wifi</b> .	
System	Then I recommend the a and b guest house. Would you like me to book you a room?	
User	Does that also have free parking available?	
System	No, it doesn't. should I recommend you a place with free parking instead?	
User	No, but I am looking for a particular restaurant. Its name is called <b>bangkok city</b>	
System	Bangkok city is an expensive Thai restaurant in the <b>centre</b> of town. They are located at 24 green street city centre. Their postcode is cb23jx. Would you like a reservation?	
User	All I needed today was the address, thank you	

Table 1: A DST example in MultiWOZ dataset

well. Our contributions are summarized as follows: (1) We show that explicit ASR error correction can improve the performance of dialogue systems with spoken corpora as input. (2) We show that post-processing can mitigate errors in words such as proper nouns. (3) We successfully construct a dialogue system that performs well with spoken utterance input.

## 2 Related Work

### 2.1 Dialogue State Tracking

Dialogue state tracking (DST) is one of the components of a task-oriented dialogue system that maps partial dialogues to the dialogue state. It usually extracts the user's goal and intent in the form of a slot-value pair through the user and system dialogue conversation. As an example in Table 1, the DST task is to extract dialogue states such as the value of *guesthouse* in the slot of *hotel-type* and the value of *bangkok city* in the slot of *restaurant-name* from user's utterance. There are several methods have recently attracted attention in DST tasks.

#### Dialogue Systems with Description Input

Some works have been proposed which include task descriptions as input, where the descriptions related to the dialog system slot or slot value examples are added as input data (Shah et al., 2019). Zhao et al. (2022) showed that a language description-driven system shows a better understanding of task specifications, higher performance on state tracking, improved data efficiency, and effective zero-shot transfer to unseen tasks.

**Prompting Dialogue System** In the NLP community, it has been shown that large language models such as GPT-3 (Brown et al., 2020) and LaMDA (Thoppilan et al., 2022) can do few-shot learning without fine-tuning. In the dialogue state tracking task, Madotto et al. (2020) applied GPT-2 by priming the model with dialogue state value pair examples. Their model works well on dialogue

state tracking with few shot examples without fine-tuning.

### 2.2 Spoken Dialogue System

In spoken dialogue systems, the discrepancy between text and speech conversations makes it difficult for text-based dialogue systems to perform well. The differences arise from different styles of spoken and written conversations, extra noise from grammatical errors, disfluencies or barge-ins, and speech recognition errors. To solve these issues, benchmark sets using audio were proposed.

Hemphill et al. (1990) and Bennett and Rudnicky (2002) proposed the speech dialogue benchmark which consists of fairly limited corpora focusing on travel reservations. Furthermore, there is no dialogue state annotation. Recently, with spoken dialog systems becoming popular, various speech dialogues benchmark sets have emerged that overcame these limitations, such as DSTC2 (Henderson et al., 2014) and DSTC10 knowledge-grounded task-oriented dialogue modeling on spoken conversations track (Kim et al., 2021). However, these benchmark sets were still insufficient to solve the speech-aware dialogue system because they had no dialog annotations, poor ASR system results, and/or a lack of audio data.

## 3 Speech-aware Dialog Systems Task

Here we provide background on the Speech-aware dialogue system technology track in DSTC11, which was introduced to overcome the challenges in implementing a good speech dialogue system with the previous speech dialog system benchmarks.

### 3.1 Dataset

The dataset combines the Multi-domain Wizard-of-Oz (MultiWOZ) dataset (Kim et al., 2019) and audio-related features. The MultiWOZ dataset

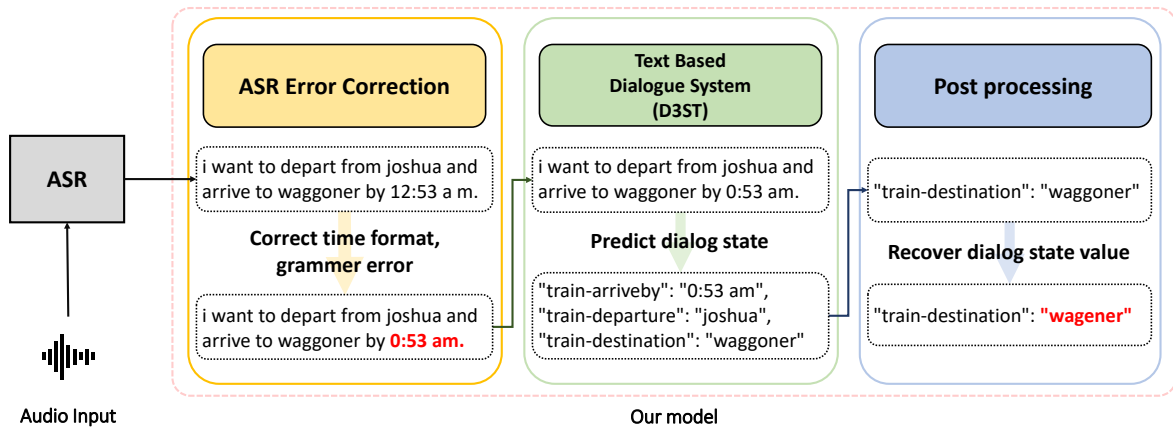


Figure 1: Our model structure

is a large-scale human-human conversational corpus spanning over seven domains, including about 10,000 multi-turn dialogues. Each dialogue is rich in annotations such as ‘goal’, ‘meta-data’, ‘dialog act’, and user and system utterances. Audio-related features are expressed in the following four kinds.

1. Raw audio in the standard PCM format, 2 bytes per sample, at 16KHz sampling rate. Given as two types: TTS and human.
2. Audio encoder output from the ASR system, consisting of 512-dimension vectors at a rate of 75 vectors per second.
3. Transcripts from the ASR system (ASR hypothesis).
4. Time alignment describing how the recognized words map to the encoder output sequences. For example, *w:while t:2 w:in t:5 w:cam t:8 w:bridge t:11 w:â- t:15 w:i t:15 ...*

The track tried to reduce the performance difference according to the ASR model by providing high-quality ASR results and feature vectors.

In addition, the state value of the dataset has been updated because the state value of the original MultiWOZ dataset was overlapped in the dev and test set compared to the train set, so fair evaluation could not be performed. Also, by changing the slot value in the conversation, the dataset was augmented 100 times.

Lastly, we used datasets with three kinds of audio as input to evaluate the spoken dialog system. The three types of datasets are classified according to the type of audio: *TTS-Verbatim* generated by TTS, *Human-Verbatim* by human speech,

and *Human-Paraphrased* by paraphrased human speech.

### 3.2 Evaluation Methods

The evaluation metric is dialogue state tracking<sup>3</sup>, one of the base evaluation metrics of the MultiWOZ dataset. Dialogue state tracking is a primary evaluation method that intuitively checks how well the dialogue model works for audio input. Joint goal accuracy (JGA) was used as the first evaluation metric, and slot error rate (SER) and F1 score were used as the second evaluation metric.

## 4 Methods

In this section, we describe our model that participated in the speech-aware dialogue systems technology challenge track in DSTC11. Figure 1 presents our model architecture. Our system consists of three modules: (1) ASR error correction, (2) Text Based Dialogue System, and (3) Post-processing. The overview of the process followed by our model is as follows:

1. Take the given ASR result from the audio data set to create an input.
2. Insert the input data into the **ASR error correction module** to obtain the text that corrected the ASR error.
3. Output the value of the dialog state with the **dialog system** learned based on the text
4. Run the **post processing module** to recover the dialog state value

<sup>3</sup><https://github.com/jasonwu0731/trade-dst>

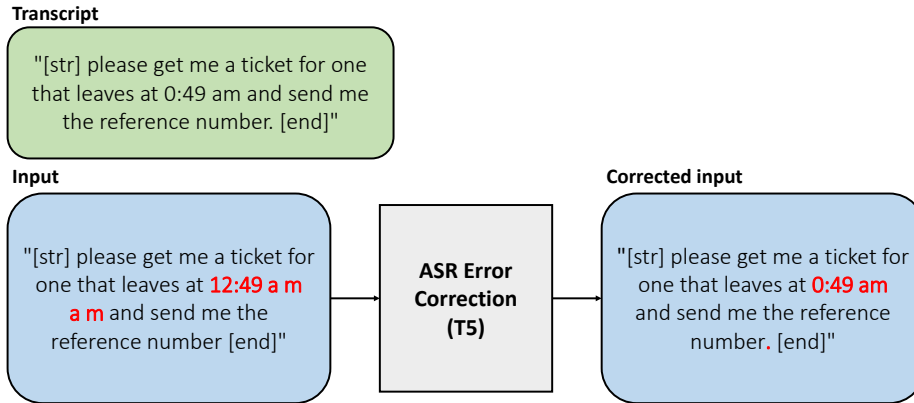


Figure 2: ASR correction example (MultiWOZ *mul0207.json*)

#### 4.1 ASR error correction

In general, we use the ASR model to transcribe audio to text. However, errors arising from the ASR module can be propagated to the dialog system that uses them as input (Dutta et al., 2022), thus it is necessary to correct these ASR errors. Since we have the ground truth text and corresponding ASR hypothesis data, we need to calibrate the ASR hypothesis to ground truth text. As in previous studies, ASR errors can be improved by applying the n-best technique (Zhu et al., 2021), but applying the n-best technique requires audio data for training, which can be very expensive. In addition, in the case of the ASR model, text converted through ASR can be inappropriate for certain downstream tasks. Therefore, instead of the n-best technique, we utilize an ASR error correction module to convert the input to a similar text in our ground truth transcript through a seq2seq-based model.

Figure 2 shows the flow of our ASR error correction module. By comparing the ground truth text and ASR hypothesis, it can be seen that there are many differences. There are special character disappearances, errors in certain formats such as time, as well as spelling errors. To fix this, we implemented a model that was trained to generate ground truth text from the ASR hypothesis based on T5 (Raffel et al., 2020). Therefore, our model can import recovered input instead of the ASR hypothesis which has a higher error compared to the original text.

#### 4.2 Inclusion of task description

Previously, for TOD systems, there have been two main approaches: a modular approach to solve each task by module and the end-to-end approach

(Madotto et al., 2018). The end-to-end approach could not show good performance due to the propagation of ASR errors in the spoken dialog system. In addition, the dataset provided on the track was difficult to perform well compared to the conventional MultiWOZ dataset.

Therefore, we used Description-Driven DST (D3ST; Zhao et al., 2022) as a baseline to increase data efficiency by utilizing the description provided by MultiWOZ<sup>4</sup>. Also, we chose a T5-based generation model that showed higher efficiency in the DST Task.

Figure 3 shows an example of the input structure of our DST model. First, we transformed the input data into D3ST format and applied an indexing mechanism, handling categorical slots. Additionally, we applied random ordering techniques to target dialogue states for each example during training to prevent dependence on the order of predicted slots. When predicting a slot using a generation-based model, this method shows that the model does not depend on the order of slots, and as a result, the slot prediction performance increases.

#### 4.3 Post Processing

There were many proper nouns such as hotel and restaurant names, but in the case of our model, which is using the ASR result as input, it was challenging for ASR error correction to correctly fix these proper nouns. In the previous study, the slot values were recovered by applying an encoder-decoder structure in a hierarchical manner (Ren et al., 2019) or modified to a similar value using the Levenshtein distance ratio (Levenshtein et al.,

<sup>4</sup><https://github.com/budzianowski/multiwoz>

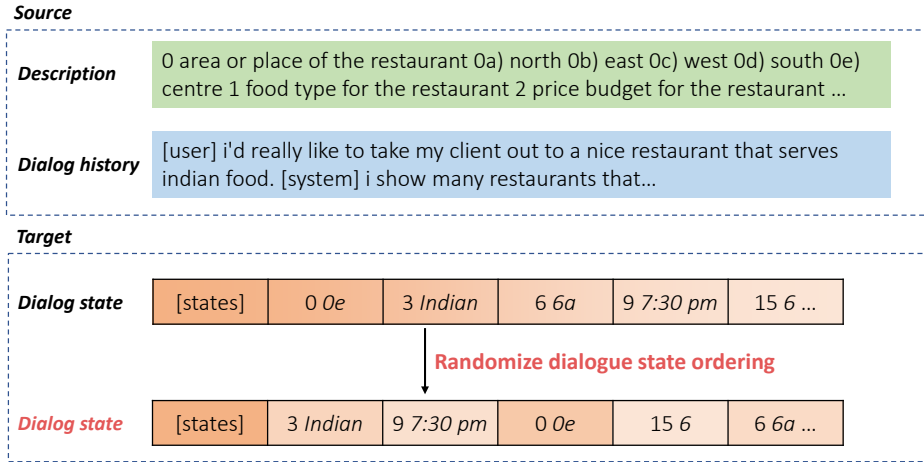


Figure 3: An example of DST model input. We applied a random ordering mechanism to D3ST-based input.

1966). To correct these proper nouns, we created a proper noun database based on Wikipedia data, and we created a module that uses word similarity to recover spelling error words.

There are various algorithms for calculating word similarity. We need to change the misrecognized word caused by ASR error to the correct word, but the spaCy library<sup>5</sup> could not be used because it calculates the similarity of cosine in the vector of the word. In addition, the Gestalt pattern matching (Ratcliff and Metzner, 1988) algorithm is not symmetric, so it shows different similarities depending on the input order, and the performance is slow. Therefore, we applied the Levenshtein distance ratio (Levenshtein et al., 1966) algorithm that calculates similarity by editing (add/delete/update).

## 5 Experiments

We trained two T5 (Raffel et al., 2020) models for ASR error correction and the description dialogue system using huggingface. T5-base(220M) was used for ASR error correction and T5-base, large(770M) for the description dialog system. We used pre-trained weights by Wolf et al. (2020) and tokenized each sentence into sub-words using T5Tokenizer<sup>6</sup>.

For fine-tuning, we used the AdamW optimizer (Loshchilov and Hutter, 2017) with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , weight decay = 0.01 and a learning rate of 0.0001. We used a batch size of 8, an input length of 256, the output length of 256 for ASR error correction, and batch size of 6 (3 for T5-large),

<sup>5</sup><https://spacy.io/api/token#similarity>

<sup>6</sup><https://github.com/huggingface/transformers>

an input length of 1024, and an output length of 512 for the description dialogue system.

### 5.1 ASR error correction Result

Post Processing	ASR	ASR correction
NONE	92.4%	<b>57.5%</b>
REMOVING SPECIAL CHARACTER	58.7%	<b>22.0%</b>

Table 2: Comparison of sentence error rate between the text of ASR without post-processing and of our ASR correction method. The sentence error rate is the number of incorrect sentences divided by the total number of sentences.

We compared the text enhanced by our ASR correction method to the text generated by ASR without correction. The result is summarized in Table 2. The first row shows the result without any post-processing text while the second row shows the result of removing special characters in the generated text. The sentence error rate is the number of incorrect sentences divided by the total number of sentences. We observe that our ASR correction method yields dramatic performance improvement, which is about a 35% increase in sentence error rate in both cases. This is because applying our ASR correction method makes the sentence from ASR more natural. Qualitative results are presented in Figure 2, which supports our claim.

### 5.2 Main Results

Table 3 shows the results of the experiment. This result is generated by training with *original transcript* and evaluating it with *tts-verbatim* of the validation set. The track organizers presented a baseline based on the D3ST model and showed dif-

	Model + Feature	JGA	SER
Baseline	D3ST (T5-base)	20.0	-
	D3ST (T5-large)	21.1	-
	D3ST (T5-large) + <b>D</b>	28.4	-
	D3ST (T5-XXL) + <b>D</b>	33.7	-
Ours	UBAR (gpt-2)	19.2	-
	D3ST (T5-base)	23.3	31.7
	D3ST (T5-base) + <b>A</b>	28.7	25.9
	D3ST (T5-base) + <b>R</b>	26.2	-
	D3ST (T5-base) + <b>R</b> + <b>P</b>	34.1	23.4
	D3ST (T5-base) + <b>R</b> + <b>P</b> + <b>D</b>	40.3	19.0
	D3ST (T5-base) + <b>R</b> + <b>P</b> + <b>D</b> + <b>A</b>	<b>41.6</b>	<b>18.3</b>

Table 3: The experiment results were trained on the original transcript and evaluated on the validation set. **A**: ASR Error Correction, **R**: Random Ordering, **P**: Post Processing, **D**: Data Augmentation

ferences in performance depending on the size of the backbone model and data augmentation.

Considering that the UBAR(Yang et al., 2021) model operates in an end-to-end fashion, we implemented our backbone model based on D3ST(Zhao et al., 2022) which can effectively solve DST Tasks. On top of our backbone model, we carefully added each feature one by one to gradually increase the performance and identify its impact.

*Random ordering* module allows us to avoid dependence on the order of slots predicted in the inference process, and *Post Processing* module corrects prediction errors in words such as proper nouns. *ASR Error Correction* module corrected ASR errors to text similar to the original transcript. Finally, using the *Data Augmentation*, we were able to create high-performance models.

Model + Feature	Pretrain. Model (# Params.)	JGA	SER
D3ST	T5-base (220M)	41.6	<b>18.3</b>
+ <b>R</b> + <b>P</b> + <b>D</b> + <b>A</b>	T5-large (770M)	<b>42.4</b>	19.0

Table 4: Comparison of performance according to model size. **A**: ASR Error Correction, **R**: Random Ordering, **P**: Post Processing, **D**: Data Augmentation

In addition, we conducted a performance comparison experiment according to the size of the T5 model. Table 4 shows the result of the experiment. The performance of our model increased as the size of the base model increased, and Zhao et al. (2022) also showed that the performance difference between the XXL model and the large model was very large for the D3ST model. It would have been nice if we had experimented with a huge model such as XXL, but due to experimental constraints, we could only experiment with the large model.

Finally, we submitted our final model based on the large model.

Table 5 is the official results of the test submission by the participants. A total of 6 teams submitted, and each team could submit up to 2 systems, so a total of 11 systems were submitted. We submitted a model that recorded 42.4 in the validation set. Finally, our model achieved third place, with JGA 40.2 for *tts-verbatim* in the challenge.

In particular, our model showed a small gap between *human-verbatim* and *human-paraphrased* data because it was implemented through the *ASR Error Correction* module in a flexible form for sentence structure.

### 5.3 Worst-Case Analysis

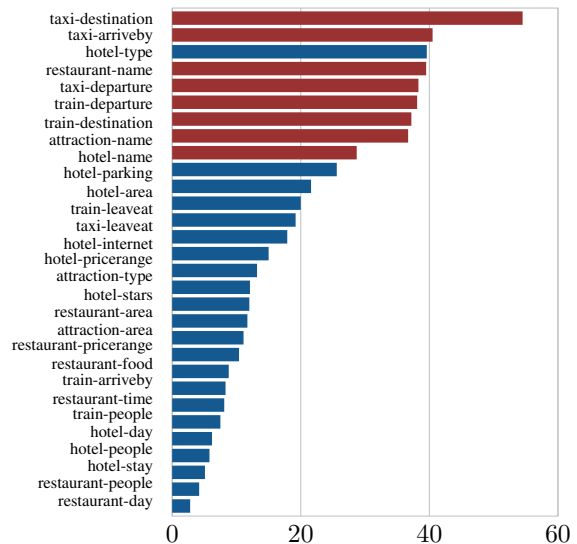


Figure 4: Slots error rate per each slot. Most slots with high slot error rates are slots with proper nouns as slot values, for example, *taxi-destination* (54.5%), *taxi-arriveby* (40.5%), and *restaurant-name* (39.5%). Red-colored slots contain proper nouns.

We hypothesize several approaches that potentially improve the performance of our model. Figure 4 shows the distribution of slot error rates for each slot. Slots containing proper nouns, such as *hotel-name* and *restaurant-name*, showed high slot error rates. Like DSTC11-Track3, in the case of a Dialogue system that uses ASR results as input, there can be many problems with proper nouns. We tried to correct these errors through post-processing, but there was still a lot of ambiguity because we have no dataset or ontology for proper nouns in the model. So we think that defining the appropriate ontology of the system

System	Joint Goal Accuracy			Slot Error Rate		
	TTS-verbatim	Human-verbatim	Human-paraphrased	TTS-verbatim	Human-verbatim	Human-paraphrased
F-p	44.0	39.5	37.9	17.1	20.0	20.4
F-s	40.4	36.1	34.3	19.2	21.9	22.4
<b>C-p (ours)</b>	<b>40.2</b>	<b>31.9</b>	<b>31.8</b>	<b>20.9</b>	<b>28.1</b>	<b>27.2</b>
A-s	37.7	30.1	30.7	20.3	26.9	26.2
C-s	33.1	28.6	28.1	25.0	28.7	29.5
D-s	30.3	23.5	23.2	26.6	36.5	35.1
B-p	27.3	23.9	22.6	26.2	30.0	30.6
D-p	30.3	23.5	23.2	28.0	36.7	36.0
A-p	21.9	21.2	20.0	32.8	33.5	33.8
B-s	22.4	19.2	18.3	28.7	32.2	32.6
E-p	21.3	20.0	18.2	35.1	35.5	35.3

Table 5: Official results for test submissions by DSTC11–Track3. The text in bold indicates our model. The first letter of System means **Team**, and the second letter means System, **p** is **Primary System** and **s** is **Secondary System**.

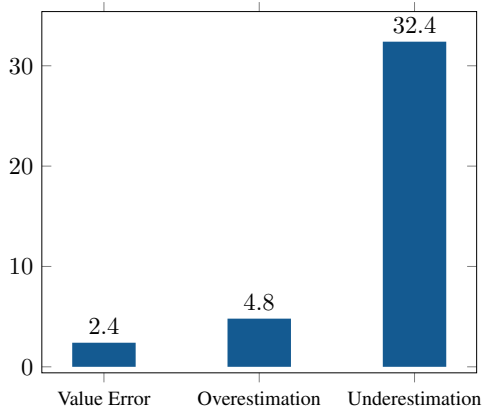


Figure 5: The cause of the error in *hotel-type* slot. Most of the reasons for the error in *hotel-type* slot were believed to have been caused by underestimation.

and post-processing will lead to even better performance.

Figure 5 shows the distribution of the error cause of the *hotel-type* slot, which showed a high slot error rate even though the slot does not include proper nouns. As a result of the analysis, most of the errors appeared to be caused by underestimation. We believe that this is caused by many cases in which hotel types are included in the hotel name, such as *guest house* and *hotel*. Therefore, we think that it would be good to solve the inference about the hotel type by separating the hotel name and the hotel type.

## 6 Conclusion

In this paper, we proposed an ASR dialogue pipeline model to solve the speech-aware dialogue system technology track in DSTC11. First, we reduce the gap when applying spoken data to text-based dialogue systems by modifying ASR results

that differ from the original text such as time formats or special characters through the ASR error correction module. Secondly, the text-based dialogue system, a model including descriptions with several techniques, was able to get good performance even with low input data feature quality of audio data. Lastly, by recovering errors such as proper nouns that cannot be caught in ASR error correction through post-processing, a dialogue system that produces high performance even on the speech database was created. In this way, we have created a model that can overcome the limitations of speech in several parts, and in the case of T5, although the model is very sensitive to parameter size, we can confirm that it is a model with high potential for development by performing well even in a small size model.

## Acknowledgments and Disclosure of Funding

This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government (22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System), Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2020-0-00940, Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

## References

- Christina Bennett and Alexander Rudnicky. 2002. The carnegie mellon communicator corpus.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Samrat Dutta, Shreyansh Jain, Ayush Maheshwari, Ganesh Ramakrishnan, and Preethi Jyothi. 2022. Error correction in asr using sequence-to-sequence models. *arXiv preprint arXiv:2202.01157*.
- Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.
- Omar Zia Khan, Jean-Philippe Robichaud, Paul A Crook, and Ruhi Sarikaya. 2015. Hypotheses ranking and state tracking for a multi-domain dialog system using multiple asr alternates. In *Sixteenth Annual Conference of the International Speech Communication Association*.
- Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papanagelis, Karthik Gopalakrishnan, Behnam Hedayatnia, and Dilek Hakkani-Tür. 2021. “how robust ru?”: Evaluating task-oriented dialogue systems on spoken conversations. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1147–1154. IEEE.
- Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sangwoo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.
- Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.
- Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- John W Ratcliff and David E Metzener. 1988. Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, 13(7):46.
- Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. *arXiv preprint arXiv:1909.00754*.
- Darsh J Shah, Raghav Gupta, Amir A Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. *arXiv preprint arXiv:1906.06870*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Xin Tian, Xinxian Huang, Dongfeng He, Yingzhan Lin, Siqi Bao, Huang He, Liankai Huang, Qiang Ju, Xiyuan Zhang, Jian Xie, et al. 2021. Tod-da: Towards boosting the robustness of task-oriented dialogue modeling on spoken conversations. *arXiv preprint arXiv:2112.12441*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.
- Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.
- Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *arXiv preprint arXiv:2201.08904*.
- Linchen Zhu, Wenjie Liu, Linquan Liu, and Edward Lin. 2021. Improving asr error correction using n-best hypotheses. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 83–89. IEEE.