CopyT5: Copy Mechanism and Post-Trained T5 for Speech-Aware Dialogue State Tracking System

Cheon-Young Park*
Korea Telecom (KT)
park.cheonyoung@kt.com

Yewon Jeong*
Korea Telecom (KT)
yewon.jeong@kt.com

Eunji Ha
Korea Telecom (KT)
eunji.ha@kt.com

Haeun Yu
Korea Telecom (KT)
haeun.yu@kt.com

Chiyoung Kim
Korea Telecom (KT)
chi-young.kim@kt.com

Joo-won Sung
Korea Telecom (KT)
jwsung@kt.com

Abstract

In a real-world environment, Dialogue State Tracking (DST) should use speech recognition results to perform tasks. However, most existing DST research has been conducted in text-based environments. This study aims to build a model that efficiently performs Automatic Speech Recognition-based DST. To operate robustly against speech noise, we proposed CopyT5, which adopts a copy mechanism, and trained the model using augmented data including speech noise. Furthermore, CopyT5 performed post-training using the masked language modeling method with the MultiWOZ dataset in T5 in order to learn the dialogue context better. The copy mechanism also mitigated named entity errors that may occur during DST generation. Experiments confirmed that data augmentation, post-training, and the copy mechanism effectively improve DST performance.

1 Introduction

Task-oriented dialogue systems are used in various fields and are intimately connected to our daily lives. They can help users perform tasks that are frequently encountered in daily life, such as restaurant reservations and train ticket reservations. However, as most of these dialogue systems are implemented using a text-based dialogue corpus, they are very weak when implemented in actual speech interface-based dialogue systems.

In actual human speech, Automatic Speech Recognition (ASR) data contains errors generated because of similar words, inaccurate pronunciation, and noise in the environment being considered, making it difficult to use it directly for learning conversation system models. Various attempts have been proposed to implement a robust speech interface-based dialogue system, such as Fazel-Zarandi et al. (2019) and Liu et al. (2021).

In Track 3 of the DSTC11 summer track, we aim to secure a dialogue system in consideration of speech recognition environments that include speech noise and paraphrasing. Accordingly, we propose a dialogue system based on voice audio and evaluate our model based on a given dataset.

We adopt simpleTOD (Hosseini-Asl et al., 2020) in an end-to-end manner, which is robust in noisy-labeled annotation among task-oriented dialogue systems. To secure a dialogue corpus of various expressions, CoCo (Li et al., 2021) and LAUG (Liu et al., 2021) toolkit were applied to augment the text corpus, and post-training (Han et al., 2022) was performed to better understand the secured text data. Furthermore, in order to improve the generation error occurring in the normal DST model, we propose a dialogue system that applies an effective copy mechanism (See et al., 2017) to out-of-vocabulary resolution and applies certain efficient post-processing techniques.

2 Related Work

2.1 SimpleTOD

A typical task-oriented dialogue system comprises three tasks: Natural Language Understanding (NLU), Dialog State Tracker (DST), and Natural Language Generation (NLG), and suggests an appropriate dialogue system for each sub-task. However, simpleTOD (Hosseini-Asl et al., 2020) proposes a casual language model that can encompass all sub-tasks. Through a simple approach to recast to a single sequence prediction problem, it achieved state-of-the-art performance in the DST task domain and confirmed that the performance impact was low even in noisy-labeled annotations. We adopted the simpleTOD model in an end-to-end manner to secure a robust dialogue system against noise errors, which is the goal of Track 3 of DSTC11 summer track.

DST aims to predict the previous dialogue con-

*Equal Contribution.

Proceedings of The Eleventh Dialog System Technology Challenge, pages 89–94

text and current dialogue states. In general, the next dialogue is predicted from the first utterance of the dialogue to the previous utterance at every dialogue turn. However, an issue arises in that it reacts sensitively to the length of dialogue in predicting the current dialogue state. As the dialogue length increases, the initial dialogue acts as noise, whereas for short dialogue lengths, there is insufficient information for prediction in the previous dialogue. Yang et al. (2021) tested the effect of context information of varying granularity on DST. Accordingly, we set the maximum value of dialogue history so that consistent and universal dialogue tracking can be conducted.

2.2 Copy Mechanism

Copy mechanism is a useful method for maintaining the context of the input sentence and solving the out-of-vocabulary problem. It improves performance by mitigating the out-of-vocabulary problem by outputting the words of the input sentences that appear less frequently in tasks from the decoder, such as machine translations (Zhang et al., 2021) and summarization (See et al., 2017). In DST, the copy mechanism is effective in inferring the dialogue state from the long-distance dialogue history (Wu et al., 2020). Based on previous research, in this study, a copy mechanism was applied to T5 to solve out-of-vocabulary problems such as entity name and to maintain the context of the input dialogue.

2.3 Data Augmentation

Data augmentation refers to generating or transforming data in order to supplement the training data. Liu et al. (2021) states that a dialogue system trained on text-based data is not robust in an audio-based data environment containing noise. CoCo (Li et al., 2021) is a method of generating a new dialogue turn by generating user utterances from system utterances and dialogues states. CoCo improved the performance of the DST model by creating a domain slot combination with a low frequency of appearance among the dialogue states of the MultiWOZ data. CopyT5 was trained LAUG and CoCo data to operate robustly on data containing speech noise and paraphrased data.

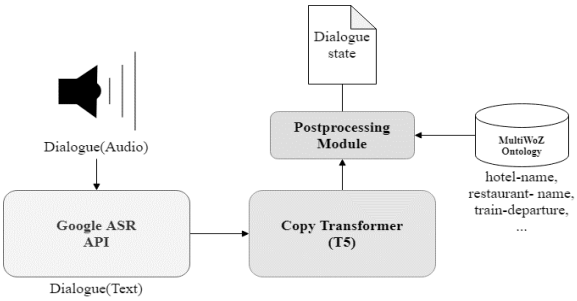

Figure 1: Overview of DST inference system.

3 Method

3.1 Overview

Figure 1 show the inference flow of our system. In the training stage, we used the training dataset provided by DSTC11 Track 3 and augmented dataset. However, in the test phase, the provided Raw Audio Data was recognized as Google ASR API[1] to reduce the noise of conversation data. Google ASR API recognizes time expression and entity name expression better, and errors in transcription are relatively low.

3.2 Post-Training

To better understand the MultiWOZ dataset, CopyT5 used post-training. Post-training means that a model is trained on dataset from the same domain before the fine-tuning stage. We train the T5 model on MultiWOZ using Masked Language Modeling (MLM) loss($\mathcal{L}_{MLM}$).

$$\mathcal{L}_{MLM} = -\sum_{j \in C} logP(y_j|x;\theta) \quad (1)$$

To learn the dialogue context better, the masked part of the dialogue history (of up to three turns) was generated. In Equation 1, j denotes the position of the masked word, C denotes the set of masked words, and y_j is the masked word. x is the input sequence, and θ is the model's parameter.

3.3 Granularity

In DST, granularity refers to the maximum number of turns to consider as current dialogue history. Dialogue has a co-reference in which the utterance of previous turns is related to the current turn. However, if all previous turns are input to the model, the input sequence becomes long and can become noise in the DST model. In contrast, if only the utterance of the current turn is used as the input

[1] https://cloud.google.com/speech-to-text

sequence to reduce the noise, the co-reference of the previous utterances cannot be identified.

Therefore, in this study, up to three turns of dialogue are considered as dialogue history, and the previous turns are replaced with the dialogue state.

3.4 Post-Processing

The generative DST model may give an error while generating dialogue states. Therefore, if the model generated a domain-slot that does not exist in the MultiWOZ dataset, it was removed, and some incorrect domain-slots were revised or deleted as per rules.

Most domain-slot information such as 'time', 'day', and 'area' slots appears in the current turn utterance. Especially in the case of 'day' and 'area' slot, the values are categorical, so we extract and reflect them in the results via a rule-based method.

ASR results may not be accurate for named entities such as restaurant names, hotel names, and station names. Consequently, we built an ontology using MultiWOZ and the named entities collected from the web. We calculated the similarity between the name-related domain-slot values inferred by the model and named entities in the ontology. If the similarity was 0.9 or higher, the value was substituted with the named entity in the ontology.

3.5 Model Architecture

In the generative DST model, an error may occur in the process of generating the named entity in the dialogue. To reduce this error, CopyT5 adopts a copy mechanism. The copy mechanism increases the generation probability of the named entity in the input, so it helps the model to output the named entity as it is in the input. Figure 2 illustrates the architecture of CopyT5.

$$h = T5Encoder(x; \theta) \tag{2}$$

$$s = T5Decoder(h, y; \theta) \tag{3}$$

$$e = v^T gelu(W_h h + W_s s) \tag{4}$$

$$a = softmax(e) \tag{5}$$

Attention a is calculated as given in Equation (5). x is the dialogue history, i.e., input sequence of the encoder, and y is the gold dialogue state, i.e., input sequence of the decoder. h is the output state of the encoder, and s is the output state of the decoder. v and W_h,W_s are learnable parameters.

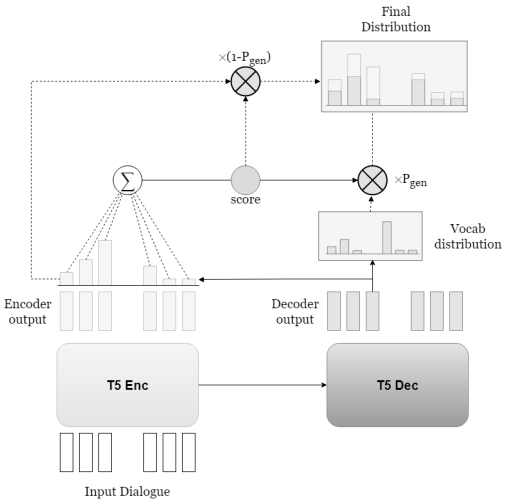

Figure 2: Overview of CopyT5.

$$h^* = \sum a * h \tag{6}$$

The context vector h^* is calculated by using the attention score as given in Equation (6).

$$p_{vocab} = softmax(V_s s + V_h h^*) \tag{7}$$

p_{vocab} is the probability of the entire vocabulary that the decoder can generate. p_{vocab} is calculated as in Equation (7), and V_s, V_h is the learnable parameter.

$$p_{gen} = \sigma(w_h h^* + W_s s + w_x x) \tag{8}$$

p_{gen} is a value that adjusts the probability of words in the input sequence and the words generated by the decoder. In Equation (8), x denotes the previous state of the decoder. As the transformer calculates the decoder state in parallel during the training process, x is the same as the value shifted to the right of the decoder output. w_h, w_s, w_x is the trainable parameter.

$$P(w) = p_{gen} p_{vocab}(w) + (1 - p_{gen})a \tag{9}$$

The final output of CopyT5 is as shown in Equation (9). CopyT5 determines copy probability and generation probability according to the value of p_{gen}. $P(w)$ is probability of predicted word w.

4 Experiments and Results

4.1 Dataset

To train our models, we used several datasets including the DSTC11 training dataset. We also used

Model	Training Datasets				
	MultiWOZ 2.1			CoCo	LAUG
	Cleaned text	DSTC11 transcription	Huggingface speech2text		
CopyT5 (Base)	56,778	56,778	56,778	22,471	-
CopyT5 (Large)	56,778	56,778	56,778	22,471	36,000

Table 1: The number of turns in training datasets for each model

System	TTS Verbatim	Human Verbatim	Human Paraphrased
F-p	44.0	39.5	37.9
F-s	40.4	36.1	34.3
C-p	40.2	31.9	31.8
A-s	37.7	30.1	30.7
C-s	33.1	28.6	28.1
D-s	**30.3**	**23.5**	**23.2**
B-p	27.3	23.9	22.6
D-p	**28.6**	**21.8**	**21.4**
A-p	21.9	21.2	20.0
B-s	22.4	19.2	18.3
E-p	21.3	20.0	18.2

Table 2: Overall JGA results of DSTC11 Track 3.

System	TTS Verbatim	Human Verbatim	Human Paraphrased
F-p	17.1	20.0	20.4
F-s	19.2	21.9	22.4
A-s	20.3	26.9	26.2
C-p	20.9	28.1	27.2
C-s	25.0	28.7	29.5
B-p	26.2	30.0	30.6
B-s	28.7	32.2	32.6
A-p	32.8	33.5	33.8
D-s	**26.6**	**36.5**	**35.1**
E-p	35.1	35.5	35.3
D-p	**28.0**	**36.7**	**36.0**

Table 3: Overall SER results of DSTC11 Track 3.

our own augmented data. To make our models robust to various ASR errors and familiar with the ASR data, we augmented the training dataset with HuggingFace speech2text transcript and LAUG. We also used CoCo to augment the training data to help our models better predict proper noun values. Table 1 summarizes the number of turns of training data to train our models.

4.2 Training Details

We used two T5 models (Raffel et al., 2020) with different sizes (base, large). We also conducted post-training in the MultiWOZ domain using MLM loss. The model is trained over five epochs in the post-training stage. At the fine-tuning stage, we trained the model for up to 30 epochs and used early stopping when the performance on the dev dataset was the best. We used two GPUs with 16 batch sizes for training the base model, and four GPUs with four batch sizes for training the large model. For the other hyper-parameters, we used the default hyper-parameters setting of the HuggingFace T5 model[2].

4.3 Evaluation Metrics

We used two metrics to evaluate our models. Joint Goal Accuracy (JGA) is used as the main metric. JGA is widely used to evaluate DST models. At each turn, JGA is 1 only if all domain-slot and value pairs are predicted correctly; otherwise it is 0. This is quite strict and the model gets the worst JGA when it is wrong at the earlier turns in the dialogue. Slot Error Rate (SER) is used as a secondary metric. SER is the ratio of total number of slot errors (substitutions + deletions + insertions) and total number of slots in reference across all the dialogues.

4.4 Results

The base-sized T5 model is additionally post-trained in the MultiWOZ domain using MLM loss. However, it is not effective in the large-sized T5 model, so we used the the open-source T5 large model as it is. Tables 2 and 3 present the JGA and SER results of submissions for Track 3 of DSTC11 respectively. D-s is the base-sized CopyT5 model that conducted post-training in the MultiWOZ domain using MLM. D-p is the large-sized CopyT5 model for which post-training is ineffective, so we only conducted fine-tuning to D-p.

In terms of Slot Error Rate, our systems ranked low. However, the JGA score was ranked relatively high compare with Slot Error Rate score of our system. The CopyT5 model over-generated the 'restaurant-name', 'hotel-name', and 'attraction-

[2] https://huggingface.co/models

System	TTS Verbatim	Human Verbatim	Human Paraphrased
Base (a)	24.43	19.94	19.54
Base (b)	27.06	20.74	21.03
Base (c)	27.34	21.08	20.97
Base (d)*	30.3	23.5	23.2
Large (a)	26.33	21.50	21.32
Large (b)	23.71	18.82	18.67
Large (c)**	28.6	21.8	21.4
Large (d)	27.33	21.84	20.79

Table 4: Overall JGA results of ablation study.

System	TTS Verbatim	Human Verbatim	Human Paraphrased
Base (a)	31.83	39.72	39.29
Base (b)	28.86	38.76	36.92
Base (c)	30.97	42.47	40.76
Base (d)*	26.6	36.5	35.1
Large (a)	30.32	38.42	37.48
Large (b)	33.84	42.49	41.39
Large (c)**	28.0	36.7	36.0
Large (d)	30.59	37.94	38.24

Table 5: Overall SER results of ablation study.

name' domain-slots. We believe that the CopyT5 based DST model was trained to over-extract values for name-related named entities in dialogue history. In contrast, time-related domain-slots such as 'train-leaveat' and 'train-arriveby' were generated less often because the ASR results contained considerable noise in time-related expressions.

To prove the effectiveness of our approach, we conducted an ablation study. The list of ablation studies is as follows:

(a) simpleTOD DST model using T5 without data augmentation

(b) CopyT5 without data augmentation

(c) CopyT5 with data augmentation

(d) CopyT5 with data augmentation, post-training

Tables 4 and 5 summarize the performance of various models (* is D-s, ** is D-p). First, in the case of the base-sized model, when comparing (a) and (b), we can observe up to 2.63% increase in JGA by using the copy mechanism. Furthermore, comparing (b) and (c), JGA increased by up to 0.34% using copy mechanism and data augmentation together. The base-sized CopyT5 with data augmentation and post-processing gives the best performance. Compared to (c), (d) showed significant improvement in JGA by up to 2.96%. In contrast, in the case of the large-sized model, when the copy mechanism was used without data augmentation (comparing (a) and (b)), the performance deteriorated. If the copy mechanism is used, the number of learning parameters of the model increases. Thus, in the large-size model, we believe that the data is not enough to train the model sufficiently. Comparing (b) and (c), we can observe significant improvements in JGA up to 4.89% when we used CopyT5 with data augmentation. We also observe that post-training was not effective in the large-sized CopyT5 by comparing (c) and (d).

5 Conclusion

In this study, DST was performed using the CopyT5 model to which the copy mechanism was applied. The copy mechanism contributed to performance improvement by suitably extracting object names that appeared in the conversation. In addition, post-training helped CopyT5 improve performance in the DST fine-tuning step by pre-learning conversational domain data. However, in the large-sized CopyT5 model, the copy mechanism and post-training were not effective, but the performance improved as a result of applying data augmentation together.

Limitations

The Copy Mechanism helped improve the performance of JGA, but suffers from excessive extraction of domain-slot values. In addition, by applying the copy mechanism, the number of learning parameters increased, and the DST performance decreased despite learning the same training dataset in a large-sized model.

In the future, we will experiment with methods such as data augmentation and prompt learning to efficiently learn the increased learning parameters by applying additional mechanisms in large-scale language models.

References

Maryam Fazel-Zarandi, Longshaokan Wang, Aditya Tiwari, and Spyros Matsoukas. 2019. Investigation of error simulation techniques for learning dialog policies for conversational error recovery.

Janghoon Han, Joongbo Shin, Hosung Song, Hyunjik Jo, Gyeonghun Kim, Yireun Kim, and Stanley Jungkyu

Choi. 2022. External knowledge selection with weighted negative sampling in knowledge-grounded task-oriented dialogue systems.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

SHIYANG Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness testing of language understanding in task-oriented dialog. 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 2467–2480, Online. 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(1).

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Peng Wu, Bowei Zou, Ridong Jiang, and AiTi Aw. 2020. GCDST: A graph-based and copy-augmented multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1063–1073, Online. Association for Computational Linguistics.

Puhai Yang, Heyan Huang, and Xian-Ling Mao. 2021. Comprehensive study: How the context information of different granularity affects dialogue state tracking? 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 2481–2491, Online. Association for Computational Linguistics.

Tong Zhang, Long Zhang, Wei Ye, Bo Li, Jinan Sun, Xiaoyu Zhu, Wen Zhao, and Shikun Zhang. 2021. Point, disambiguate and copy: Incorporating bilingual dictionaries for neural machine translation. 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 3970–3979, Online. Association for Computational Linguistics.