Speak & Spell: LLM-Driven Controllable Phonetic Error Augmentation for Robust Dialogue State Tracking

Jihyun Lee[1], **Solee Im**[3*], **Wonjun Lee**[2], **Gary Geunbae Lee**[1,2]
[1]Graduate School of Artificial Intelligence, POSTECH, Republic of Korea
[2]Department of Computer Science and Engineering, POSTECH, Republic of Korea
[3]KT
{jihyunlee, lee1jun, gblee}@postech.ac.kr
{solee.im}@kt.com

Abstract

Dialogue State Tracking (DST) is a key part of task-oriented dialogue systems, identifying important information in conversations. However, its accuracy drops significantly in spoken dialogue environments due to named entity errors from Automatic Speech Recognition (ASR) systems. We introduce a simple yet effective data augmentation method that targets those entities to improve the robustness of DST model. Our novel method can control the placement of errors using keyword-highlighted prompts while introducing phonetically similar errors. As a result, our method generated sufficient error patterns on keywords, leading to improved accuracy in noised and low-accuracy ASR environments.

1 Introduction

Task-oriented dialogue systems (TODs) assist users in achieving specific objectives through conversations and are used in various sectors, including customer service and hotel reservations. A crucial component of these systems is Dialogue State Tracking (DST), which extracts vital information from conversations in a slot-value format (e.g., hotel-name: Claire Hotel). This information is essential for querying databases and generating responses (Young et al., 2013).

However, DST models face significant challenges in spoken dialogue environments, where user utterances are converted into text by automatic speech recognition (ASR) (Pal et al., 2020; Kim et al., 2021; Yoon et al., 2023). Notably, Soltau et al. (2022) observed a drastic reduction in model accuracy from 41.6% to 23.6% in such environments. This decline is primarily due to ASR errors, which frequently misrecognize named entities—a key target in DST (Nechaev et al., 2021).

To address ASR inaccuracies, data augmentation has emerged as a viable, cost-efficient strategy. Existing text augmentation methods, such as word swapping (Wei and Zou, 2019) and back translation (Sennrich et al., 2015), do not maintain audio similarity with the original text, leading to discrepancies with ASR error patterns. To bridge this gap, Sharma et al. (2020) and Jacqmin et al. (2023) synthesized audio from text with text-to-speech (TTS) model (Shen et al., 2018) and processed it through ASR, while Hrinchuk et al. (2020) and Zhang et al. (2021) employed translation model structure to introduce ASR-like errors directly into texts. Huang and Chen (2020) further leveraged word confusion network (WCN) representations for data augmentation to generate acoustically consistent errors.

Despite these advancements, prior methods often fail to provide sufficient error for DST model training. Accurately identifying key terms is vital for DST performance; thus, models need to be trained on a broad spectrum of ASR-errored keywords. Unfortunately, many current strategies do not ensure that errors are positioned within critical keywords, often generating trivial examples by altering non-essential words such as random words (Wei and Zou, 2019) or sentence structure (Sennrich et al., 2015). This oversight results in sub-optimal performance against ASR errors.

To address these limitations, we introduce Error Positioning Augmentation (EPA), a straightforward yet effective method that ensures sufficient errors in keywords. Our method leverages large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Zhang et al., 2022), which have demonstrated impressive capabilities in semantic augmentation (Whitehouse et al., 2023; Sahu et al., 2023) and precise text generation control (Sun et al., 2023; Liang et al., 2024). Despite their strengths, LLMs' potential for phonetic augmentation remains largely unexplored.

In our method, we utilize in-context learning (Brown et al., 2020) with phonetically similar examples to introduce general ASR errors and de-

[*] This work was conducted at POSTECH.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 99–111
December 20-24, 2025 ©2025 Association for Computational Linguistics

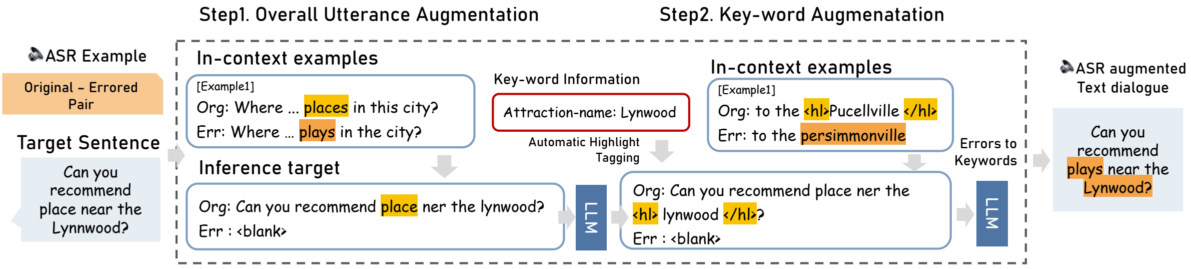
Step1. Overall Utterance Augmentation
Step2. Key-word Augmenatation
ASR Example
Original – Errored Pair
Target Sentence
Can you recommend place near the Lynnwood?
In-context examples
[Example1]
Org: Where ... places in this city?
Err: Where ... plays in the city?
Inference target
Org: Can you recommend place ner the lynwood?
Err : <blank>
LLM
Key-word Information
Attraction-name: Lynwood
Automatic Highlight Tagging
In-context examples
[Example1]
Org: to the <hl>Pucellville </hl>
Err: to the persimmonville
Org: Can you recommend place ner the <hl> lynwood </hl>?
Err : <blank>
LLM
Errors to Keywords
ASR augmented Text dialogue
Can you recommend plays near the Lynwood?

Figure 1: Illustration of the EPA process. Step 1 introduces overall ASR-like distortions using in-context examples of original–errored pairs, while Step 2 applies keyword-specific phonetic corruptions guided by highlighted slot spans. Combining both steps generates realistic and diverse ASR-style errors for robust DST training.

vise a highlighting method to explicitly localize the error to a target span. Surprisingly, without requiring extensive domain-specific user speech data, a publicly available audio dataset and a small set of in-context examples (fewer than 10 samples) are sufficient to generate a wide variety of ASR-errored keywords for DST. This significantly simplifies the error generation process.

In the experiment, to reflect diverse real-world conditions, we evaluated EPA under four ASR environments: a low-accuracy ASR system, noisy audio with café and traffic background, a paraphrased input setting where users naturally rephrased transcriptions, and a high-accuracy ASR system. In these experiments, EPA significantly improved model robustness, increasing accuracy from 45.76% to 51.12% with high keyword diversity (95.4%), surpassing the previous best-performing model. Our analysis suggests that this improvement is primarily driven by keyword-level augmentation, which effectively mitigates errors in ASR-affected values.

2 Method

2.1 Notation

Before detailing each step, we first clarify the notation. Dialogue context from turn 1 to t is denoted as D_t=$\{(s_1, u_1), ..., (s_t, u_t)\}$ where s denotes for system and u for user utterance. DST model predicts the dialogue state (also called belief state) B_t given D_t. B_t is composed with slot sl and value v pairs, denoted as B_t = $\{(sl_1, v_1), ..., (sl_J, v_J)\}$, where sl_j and v_j is j-th slot name and value. J is the total number of slots.

2.2 Step 1: ASR Error for Overall Utterance

In this step, we augmented the overall utterance by introducing general ASR errors. We began by constructing example sets for in-context learning, utilizing an open-source audio dataset (Ardila et al., 2020). From this dataset, we randomly selected 300 hours of audio along with their corresponding gold transcripts (g) and transcribed the audio using an off-the-shelf ASR model (e.g., Whisper-base (Radford et al., 2022)) to obtain the erroneous transcriptions (e). We denote this example dataset as $DB = \{(g_1, e_1), \ldots, (g_I, e_I)\}$.

Next, we inject errors into u by prompting the LLM with in-context examples. We retrieved (g, e) pairs from the database (DB) based on phonetic similarity between u and g (Figure 1, Step 1). To compute phonetic similarity, we converted the characters of both u and g into phonemes using the International Phonetic Alphabet (IPA), and calculated similarity using a frequency-based retrieval [1]. After selecting the top-k (g, e) pairs, we concatenated the instruction, in-context examples, and u into a single prompt and provided it to the LLM. This process results in the overall ASR-errored user utterance, denoted as $\dot{u}$. Concretely, $\dot{u}$ can be obtained by

$$\dot{u}_t = LLM(Inst_1 \oplus (g_1, e_1) \cdots (g_k, e_k) \oplus u_t) \quad (1)$$

where $\oplus$ denotes concatenation, and we set $k = 3$ throughout our experiments. Retrieved examples are provided in Appendix A.3.

2.3 Step 2: ASR Error for Keywords

While Step 1 introduces general ASR-style errors into u, it does not ensure sufficient error diversity in keyword tokens. To construct a more effective training dataset, we explicitly generate keyword-focused ASR errors in Step 2 (Figure 1). In this step, we highlight the keywords in $\dot{u}$ using the `<hl>` tag and instruct the LLM to inject errors specifically within the highlighted spans. For the

[1] We used BM25(Robertson et al., 2009), a retrieval model based on term frequency. While neural retrievers (e.g., DPR(Karpukhin et al., 2020)) could be applied, we opted for a frequency-based method, as neural models tend to capture semantic similarity.

Idx	Method	Examples
1	Original	Tuesday, going to bailey's crossroads please.
	+EPA	Tuesday, going to baley's crossroads, peas .
1	Original	I'd like to find a vegetarian restaurant, if possible.
	+EPA	I'd hike to find a veggie tarian restroom , if possible.
3	Original	I am going to auburn.
	+EPA	I am flowing to auburng .
4	Original	Hi! Could you find me a train to loris on thursday?
	+EPA	Oh ! Could you find me a trai to lorri on thursdae ?
5	Original	Ashby is my destination.
	+EPA	Ashy's my desity .

Table 1: Examples of ASR errors from EPA.

DST task, we treat dialogue state values (v) as keywords, although the definition of a keyword may vary depending on the task. To facilitate this process, we provide a few examples that illustrate how values within <hl> tags are intended to be modified during augmentation. Given these instructions and examples, the LLM generates an augmented utterance $\ddot{u}$ that includes both general and keyword-specific ASR errors. Formally, we obtain $\ddot{u}_t$ as follows:

$$\ddot{u}_t = LLM(Inst_2 \oplus (g_0, e_0) \cdots (g_k, e_k) \oplus \dot{u}_t). \tag{2}$$

The used prompts are provided in Appendix A.1.

2.4 Examples of EPA

Table 1 shows examples of ASR errors generated by EPA. We have highlighted utterance level overall errors in yellow and keyword-specific errors in orange . For instance, in Row 1, the model introduces a keyword-level error (bailey's → baley's) as well as an additional phonetically plausible insertion (peas), simulating realistic ASR noise. Further examples can be found in Appendix A.4.

3 Experiments

3.1 Experimental Setup

Dataset. The DSTC11 dataset (Soltau et al., 2022), an audio version of MultiWOZ 2.1 (Eric et al., 2019), comprises 8,000 dialogues for training, 1,000 for validation, and 1,000 for testing. To enhance generalization, we conducted experiments across four distinct ASR environments, characterized by Word Error Rate (WER) and noise levels: (1) a low accuracy ASR model (WER > 0.03), (2) a cafe and traffic noised audio, (3) a paraphrased setting where users naturally paraphrased the transcriptions, and (4) a high accuracy ASR model.

Metrics. For overall performance evaluation, we used joint goal accuracy (JGA), which requires all slot-value pairs to match the gold label. We also reported named entity accuracy (N-acc), the average accuracy across named entity slots.

Compared methods. We compared our method with two established approaches: text-based augmentations, **AEDA** (Karimi et al., 2021), **EDA** (Wei and Zou, 2019), and Back Translation (**BT**) (Sennrich et al., 2015), and audio-aware augmentation methods, using synthesized audio (**TTS-ASR**) and translation model structure (**ASR-translation**). Lastly, we included **Olisia** (Jacqmin et al., 2023), the top-ranked method in the DSTC11 competition.

Models. For performing EPA, we used diverse types of LLMs, including GPT-3.5(Ouyang et al., 2022), LLaMA2-7B(Touvron et al., 2023) and OPT-6.7B(Zhang et al., 2022). For the DST task, we fine-tuned a T5-base(Roberts et al., 2019) model. Further details about the experimental settings are provided in Appendix B.

3.2 Robustness Improvement through EPA

EPA improves robustness. The results in Table 2 shows the effectiveness of EPA in robustness to ASR errors. Remarkably, EPA outperformed existing text-based and audio-based augmentation, showing substantial improvement in JGA and named entity accuracy. It also surpassed the previous best-performing model, Olisia, particularly in challenging environments.

Effectiveness of keyword-specific error. In Table 2, we present an ablation study to evaluate the effectiveness of keyword-level augmentation. We found that adding keyword-specific ASR errors improved DST performance across all environments and was particularly helpful in enhancing the robustness of named entity accuracy. Additional experiments, including generalization to other backbones and tasks, as well as statistical significance analysis, are provided in Appendix C.

3.3 Comparison with Error Correction Models

In Table 3, we compare our method with two ASR error correction approaches. For Error Correction Model 1, We fine-tuned a pre-trained on 300 hours of ASR-errored and gold transcription pairs (same as in Section 2.2) in a seq2seq manner, minimizing $L = -\sum_{i=1}^{N} \log P(g_i \mid e_i)$, where g_i and e_i denote the gold and errored transcriptions, respectively. Training used early stopping (patience=3) with batch size 16. For Error Cor-

Method	Features				Low-acc ASR		Noised Aud.		Paraphrased		High-acc ASR	
	Aud.	Utt-aug	Key-aug	LLM	JGA	N-Acc	JGA	N-Acc	JGA	N-Acc	JGA	N-Acc
Baseline	-	-	-	-	29.88	45.76	29.70	46.77	28.92	48.79	34.87	52.07
AEDA (Karimi et al., 2021)	-	✓	-	-	29.90	46.46	29.74	47.48	29.12	48.86	34.94	52.32
EDA (Wei and Zou, 2019)	-	✓	-	-	29.22	47.65	28.70	49.51	28.08	49.99	33.68	53.78
BT (Sennrich et al., 2015)	-	✓	-	-	31.69	49.17	31.26	50.98	29.90	51.73	36.27	54.81
TTS-ASR	✓	✓	-	-	29.94	46.34	29.99	47.37	29.08	48.88	35.07	52.03
ASR-translation	✓	✓	-	-	30.40	47.65	30.14	48.45	29.54	50.38	35.25	53.66
EPA (Opt 6.7B)	✓	✓	✓	✓	31.82	50.73	32.03	51.92	29.57	52.49	**37.05**	55.78
w/o Keyword Aug	✓	✓	-	✓	31.43	49.63	31.51	50.56	30.41	52.02	36.34	54.57
EPA (LLAMA2-7B)	✓	✓	✓	✓	31.54	**51.12**	31.55	52.27	30.10	**53.55**	36.22	55.49
w/o Keyword Aug	✓	✓	-	✓	31.12	50.33	31.44	52.07	30.01	53.49	35.70	54.90
EPA (GPT3.5)	✓	✓	✓	✓	**32.39**	**51.12**	**32.24**	**52.70**	**30.95**	53.34	36.61	**55.87**
w/o Keyword Aug	✓	✓	-	✓	31.31	50.67	31.13	52.29	30.06	52.85	35.40	55.80
Olisia (Jacqmin et al., 2023)	-	-	-	-	30.17	46.25	30.43	48.07	29.13	49.21	36.1	52.58

Table 2: Comparison of various augmentation methods in enhancing the robustness of DST models across different ASR environments. In the feature columns, "Aud." indicates the use of audio-based augmentation, "Utt-aug." denotes utterance-level text augmentation, "Key-aug." refers to keyword-specific augmentation, and "LLM" indicates whether large language models were used for error generation. All results were averaged over three seeds for better consistency.

Method	Weak-ASR		Noised-ASR		Strong-ASR	
	JGA	N-acc	JGA	N-acc	JGA	N-acc
Baseline	29.88	45.76	29.70	46.77	34.87	52.07
+ CORRECTION MODEL 1	23.30	41.50	23.96	42.86	26.68	46.14
+ CORRECTION MODEL 2	30.14	46.79	29.82	47.68	34.62	52.00
+ EPA (with GPT, ours)	**32.39**	**51.12**	**32.24**	**52.70**	**36.61**	**55.87**

Table 3: Comparison with ASR error correction models under different ASR conditions.

RECTION MODEL 2, we adopted an off-the-shelf[2] text correction model to revise typos and transcription errors. Each model corrected the test transcriptions from the weak, noised, and strong ASR systems before DST evaluation. As in the table, CORRECTION MODEL 1 underperforms the baseline, often altering named entities unnecessarily (e.g., "Grifon" → "Bristol"), likely due to limited training diversity. CORRECTION MODEL 2 yields minor improvements, while our EPA achieves the highest robustness across all ASR conditions.

3.4 Qualitative Assessment of EPA Method

Automatic evaluation. Although Table 2 confirms EPA's effectiveness, it remains unclear whether the LLM-generated augmentations truly reflect diverse, keyword-focused ASR-style errors. To this end, we perform the quality analysis based on three metrics (Table 4): the unique word increase rate, named entity change rate, and pronunciation similarity with original sentence. The results show that EPA achieves remarkable diversity in unique words (1.81×) and the highest named entity change rate (95.47%), while maintaining high pronunciation similarity (91.57%). Notably, keyword-level augmentation plays a key role in enhancing named entity variability, increasing the change rate from 68.81% to 95.47%.

Method	Uniq. Words	NE.chg [%]	Pronoun Sim.[%]
Baseline	1	-	-
AEDA (Karimi et al., 2021)	1.00×	44.29	91.57
EDA (Wei and Zou, 2019)	0.86×	70.03	61.14
BT (Sennrich et al., 2015)	1.21×	73.46	77.17
TTS-ASR	1.01×	38.84	**98.93**
Translating	0.84×	39.59	94.07
EPA	**1.81**×	**95.47**	91.57
w/o Keyword Err.	1.57×	68.81	93.14

Table 4: Assessment of EPA dataset quality: Unique word increased rate, Named entity changed rate (NE.chg), and pronunciation similarity.

Method	Text Dist.(↓)		Phoneme Dist.(↓)	
	ASR-L	ASR-B	ASR-L	ASR-B
TTS-ASR (Whisper-B)	0.030	0.048	0.056	0.061
TTS-ASR (Whisper-S)	0.030	0.048	0.070	0.077
ASR trans. (T5-base)	**0.025**	0.039	0.104	0.116
EPA (Llama2 7B)	0.218	0.123	0.204	0.256
EPA (OPT 6.7B)	0.115	0.106	0.071	0.091
EPA (GPT 3.5)	0.033	**0.010**	**0.009**	**0.007**

Table 5: Distribution distance (JSD) between Whisper Large/Base model and simulation dataset. (ASR-L=Whisper Large, ASR-B=Whisper Base).

Comparison with Authentic ASR Errors. In this analysis, we explored the similarity between simulated ASR errors and authentic ASR errors from the perspective of edit distance. Specifically, we examined the distribution of edit distances in simulated data and in errors produced by Whisper large/base models, considering both character- and

[2] urlhttps://huggingface.co/oliverguhr/spelling-correction-english-base

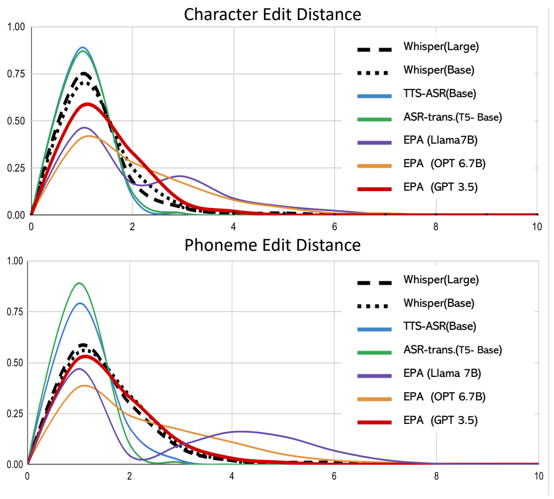

Figure 2: Distribution of edit distance. The x-axis represents edit distances, and the y-axis represents the corresponding ratio.

phoneme-level representations (Figure 2). To quantify the distributional differences, we computed the Jensen-Shannon Divergence (JSD), a symmetric variant of the Kullback-Leibler divergence (Table 5).

Our experiments yielded several noteworthy findings. Notably, the LLM-simulated errors closely matched the distribution of real ASR errors, particularly at the phoneme level. This suggests that LLMs can effectively capture pronunciation-level variations and generate realistic ASR-style distortions. In contrast, errors produced by TTS-based synthesis or ASR-translation models exhibited larger divergence from real ASR patterns, likely due to their tendency to generate more diverse but less phonetically grounded outputs.

Human evaluation. To further verify the quality of our EPA method, we conducted a human evaluation using 100 sentence pairs, each consisting of an original sentence and its augmented counterpart, with two human evaluators. Participants rated how likely the change resembled an ASR error on a 4-point Likert scale, where 1 indicated "not like an ASR error" and 4 indicated "clearly an ASR error." The average rating was 3.22 with moderate inter-rater agreement (Gwet's AC2 (Gwet, 2008) = 0.590), suggesting that most EPA-generated edits were perceived as realistic ASR errors. Details on the evaluation metric and human evaluation are provided in Appendix D.

Method	Error Type		
	Wrong	Ignore	Spurious
	Noised Audio		
Baseline	▽0% (6237)	▽0% (3654)	▽0% (2027)
EPA w/o Key-aug	▽5.29% (5907)	▽3.72% (3518)	▽**6.31**% (1899)
EPA	▽**8.19**% (5726)	▽**7.25**% (3389)	▽1.33% (2000)

Table 6: Ablation study with error analysis. **Wrong** indicates the model predicts incorrect values, **Ignore** refers to ignored mentioned slots, and **Spurious** denotes predicting values for unmentioned slots. Actual error numbers are in parentheses.

3.5 Error Analysis

We additionally analyze the impact of keyword augmentation by examining how it influences specific error types in DST predictions. Table 6 presents the percentage reduction in error rates compared to the baseline. The results demonstrate that EPA is effective in "Wrong" and "Ignore" error types, and keyword augmentation highly contributed to this improvement by decreasing the error rate from 5.29% to 8.19%. Interestingly, while keyword augmentation led to substantial reductions in "Wrong" errors, it also caused a slight increase in "Spurious" errors. This may be because the model, after repeatedly seeing phonetic noise around slot values, becomes overly sensitive and starts hallucinating unmentioned slots. A potential mitigation is to introduce an additional loss term for slot presence prediction (Heck et al., 2020; Kim et al., 2019) , helping the model better distinguish between mentioned and unmentioned slots.

4 Conclusion

We propose a novel data augmentation method tailored for DST tasks that ensures sufficient error patterns in both key phrases and overall text. By leveraging LLMs for their controlled text generation capabilities, we strategically place errors within key phrases. Our method demonstrates substantially improved robustness in DST by generating diverse, plausible keyword errors. Error case analysis reveals that keyword augmentation significantly enhances robustness against ASR errors. As the pioneering research in leveraging LLMs for generating ASR errors, we hope this work lays a strong foundation for future phonetic-based augmentation research.

Limitations

Through detailed error analysis, we identified a trade-off introduced by our keyword-focused phonetic augmentation strategy. While the augmentation helps the model become more robust to noisy slot expressions—leading to substantial reductions in "Wrong" errors—it also increases the model's sensitivity to phonetic variations. As a result, we observed cases where the model hallucinates slot values that were not actually mentioned, thereby increasing the number of "Spurious" errors. This hallucination effect represents a key limitation of our method. We attribute it to the model's repeated exposure to noisy keywords, which may cause it to overgeneralize phonetic cues as valid slot mentions. As a direction for future work, we plan to incorporate an auxiliary loss term for slot presence prediction (Heck et al., 2020; Kim et al., 2019) to help the model better distinguish between mentioned and unmentioned slots and mitigate this side effect.

Ethical Considerations

Our phonetic augmentation method, while effective for simulating ASR-style errors, may raise several ethical concerns. One such concern is the potential for accent bias, wherein phonetic transformations may disproportionately reflect majority or standard pronunciations, thereby marginalizing regional or minority accents. Another concern is the inadvertent corruption of proper names, particularly those that are less common or culturally specific, which could lead to misrepresentation or reduced inclusivity. We acknowledge these risks and emphasize that our method relies on LLMs trained on diverse and large-scale corpora. As such, the phonetic errors generated are likely to reflect dominant patterns present in mainstream ASR systems, rather than rare or region-specific variations. Nonetheless, we recognize the importance of fairness and inclusivity in language technologies and believe that future work should explore augmentation strategies that are more sensitive to accent and cultural variability.

Acknowledgements

This research was supported by the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency funded by the Ministry of Culture, Sports and Tourism in 2025 (Project No. RS-2025-02413038, 45%), the IITP-ITRC (Information Technology Research Center) Program funded by the Ministry of Science and ICT (Project No. IITP-2025-RS-2024-00437866, 45%), and the IITP Artificial Intelligence Graduate School Program (POSTECH) funded by the Ministry of Science and ICT (Project No. RS-2019-II191906, 10%).

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7252–7262, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. *arXiv preprint arXiv:2005.02877*.

Oleksii Hrinchuk, Mariya Popova, and Boris Ginsburg. 2020. Correction of automatic speech recognition with transformer sequence-to-sequence model. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7074–7078.

Chao-Wei Huang and Yun-Nung Chen. 2020. Learning asr-robust contextualized embeddings for spoken language understanding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8009–8013. IEEE.

Léo Jacqmin, Lucas Druart, Valentin Vielzeuf, Lina Maria Rojas-Barahona, Yannick Estève, and Benoît Favre. 2023. Olisia: a cascade system for spoken dialogue state tracking. *arXiv preprint arXiv:2304.11073*.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Aeda: an easier data augmentation technique for text classification. *arXiv preprint arXiv:2108.13230*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papangelis, 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 Sang-Woo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. 2024. Controlled text generation for large language model with dynamic attribute graphs. *arXiv preprint arXiv:2402.11218*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yaroslav Nechaev, Weitong Ruan, and Imre Kiss. 2021. Towards nlu model robustness to asr errors at scale. In *KDD 2021 Workshop on Data-Efficient Machine Learning*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Vaishali Pal, Fabien Guillot, Manish Shrivastava, Jean-Michel Renders, and Laurent Besacier. 2020. Modeling asr ambiguity for dialogue state tracking using word confusion networks. *arXiv preprint arXiv:2002.00768*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gaurav Sahu, Olga Vechtomova, Dzmitry Bahdanau, and Issam H Laradji. 2023. Promptmix: A class boundary augmentation method for large language model distillation. *arXiv preprint arXiv:2310.14192*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Yash Sharma, Basil Abraham, Karan Taneja, and Preethi Jyothi. 2020. Improving low resource code-switched asr using augmented code-switched tts. *arXiv preprint arXiv:2010.05549*.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, and 1 others. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Suwon Shon, Ankita Pasad, Felix Wu, Pablo Brusco, Yoav Artzi, Karen Livescu, and Kyu J Han. 2022. Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7927–7931. IEEE.

Hagen Soltau, Izhak Shafran, Mingqiu Wang, Abhinav Rastogi, Jeffrey Zhao, Ye Jia, Wei Han, Yuan Cao, and Aramys Miranda. 2022. Speech aware dialog system technology challenge (dstc11). *arXiv preprint arXiv:2212.08704*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Frederick Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating large language models on controlled generation tasks. *arXiv preprint arXiv:2310.14542*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. Llm-powered data augmentation for enhanced crosslingual performance. *arXiv preprint arXiv:2305.14288*.

Jaeseok Yoon, Seunghyun Hwang, Ran Han, Jeonguk Bang, and Kee-Eung Kim. 2023. Adapting text-based dialogue state tracker for spoken dialogues. *arXiv preprint arXiv:2308.15053*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Shuai Zhang, Jiangyan Yi, Zhengkun Tian, Ye Bai, Jianhua Tao, Xuefei Liu, and Zhengqi Wen. 2021. End-to-End Spelling Correction Conditioned on Acoustic Feature for Code-Switching Speech Recognition. In *Proc. Interspeech 2021*, pages 266–270.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

A Details of the EPA Method

A.1 Prompt Used for EPA

The prompts used in Step 1 and Step 2 are provided below.

Step 1 Prompt

Generate ASR error augmented text with similar pronunciation but different words based on the given gold text examples.
Apply character and word substitutions, additions, or deletions while maintaining the overall pronunciation and context.
Error rate should be high

. . . .

Example 1
Original: they have a single naupliar eye
ASR-errored: they have a single nor pure eye

Example 2
Original: i must have saint louis then huzza
ASR-errored: i must have st louis then hazard

Example 3
Original: i wonder uncle did not have her come
ASR-errored: i wonder uncle did not have a problem

. . . .

Now, following the above examples, generate an ASR-errored version of the following sentence:
Original: [Target utterance]
ASR-errored:

Step 2 Prompt

Change the key words in <hl> tag, to having a ASR error. ASR error has similar pronunciation with the correct word, but different charater.
Here is some example.

. . . .

Example 1
Original: I want to buy a book about <hl>luwombo best</hl> restaurant.
Keywords : luwombo best
Result: I want to buy a book about luwambo vest restaurant.

Example 2
Original: hi, i'm looking for a bus that is depart from <hl>eliot<hl/> and arriving to <hl>holiday inn williamsport<hl/>?
Keywords : eliot, holiday inn williamsport
Result: hi, i'm looking for a bus that is depart from Ellyot and arriving to holliday inn william's port

Example 3
Original: the <hl>chabuton ramen<hl/> is a restaurant on the east.
Keywords : chabuton ramen
Result: the shabuton raymond is a restaurant on the east.

. . . .

Now, following the above examples, generate an ASR-errored version of the following sentence:
Original: [Target utterance with <hl> tag]
ASR-errored:

A.2 Detailed Keyword Highlighting Strategy for EPA

Example of adding <hl> tag	
Original	Hi, I need to go to Green Day hotel, then book a table at the Grill House.
Dialogue State	hotel-name: Green Day, restaurant-name:Grill House
With <hl> tags	Hi, I need to go to <hl>Green Day</hl> hotel, then book a table at the <hl>Grill House</hl>.

Table 7: Example of keyword highlighting using <hl> tags based on dialogue state annotations.

To explicitly introduce keyword-specific ASR errors, we first identify dialogue state values from the training corpus and match them against the user utterance ($\dot{u}$). Matched values are then automatically wrapped with <hl> tags based on slot annotations (e.g., DST slot labels or NER tags), as shown in Table 7. These highlighted utterances are passed to the LLM, which is instructed to perturb the text within the <hl> tags while preserving the rest. This keyword highlighting strategy is task-agnostic and can be easily applied to other keyword-sensitive tasks such as Named Entity Recognition (NER) or Spoken Language Understanding (SLU), where certain slot values or entities are critical for downstream prediction.

A.3 Retrieved In-Context Example

In Section 2.2, we retrieved in-context examples based on phoneme-level similarity. Table 8, we present several representative examples to illustrate this retrieval process, showing how phonetically similar phrases (highlighted in color) are matched between the target and retrieved utterances. This demonstrates that the retrieval mechanism effectively captures pronunciation-level patterns relevant to ASR-style errors.

A.4 Additional Examples of ASR-style Errors

Table 9 presents additional examples of ASR-style errors generated by our EPA method, including both general and keyword-specific transformations.

B Experimental Setup

B.1 Details of the ASR Environment

- Low-acc ASR environment: Whisper-base model (74M)(Radford et al., 2022) is used for transcription. WER on LibriSpeech.test-clean is 0.05.

Example 1	
Target	can you tell me the address to the police station in point pleasant?
Retrieved 1	frayser station was not the depot on the point
+ ASR	freya station was not the deep watch on the point
Retrieved 2	can you get me the maldeamores saga
+ ASR	can you get me the melamorphos
Retrieved 3	cannot you tell her whom i am eh joseph
+ ASR	cannot you tell her whom i am
Example 2	
Target	no, i just need to make sure it's cheap. oh, and i need parking.
Retrieved 1	i need fifty ten foot long segments of wire
+ ASR	i need fifty ten foot long signals of my life
Retrieved 2	a drive with a different encoding mechanism would need different patterns
+ ASR	and drive was a different building recognition would need different patterns
Retrieved 3	to reach to calcutta you need less time to reach dhaka
+ ASR	to reach tukaukara you need last time to reach daka
Example 3	
Target	i'm open to any kind of food. i'm looking for something in the centre and on the expensive side.
Retrieved 1	kokai means open to the public or laid open
+ ASR	cook eye means open to the public all laid open
Retrieved 2	the town of beauharnois was the major centre
+ ASR	the town of bo hanwa was the major center
Retrieved 3	the gate is open at eleven
+ ASR	the gate is open at 11

Table 8: In-context examples retrieved based on phoneme-level similarity. For each target utterance (top row), we retrieve three (g, e) example pairs from the database using phonetic similarity between the target and g. Colored segments highlight phonetically similar phrases between the target and retrieved examples.

Original	**Augmented**
- no, i just need to make sure it's cheap. oh, and i need parking	- no, i just need to make sure it's sheep. oh, and i need parking.
- i am departing from marion -	- i am departing from maryland
- no, i don't need anything else right now. thank you for your assistance. good bye.	- no, i don't need anyone else right now. thank you for your persistence. good buy.
- i would like to go to sandy please.	- i would like to go to cindy please.
- i would like to keep it in the moderate range, please.	- i would like to keep it in the mod rain, please
- could i get the address for it? i would also like an expensive place to eat around it.	- could i get the actress for it? i would also like an extensive place to eat around it.
- i need to take a train out of garrett, i will be leaving town on wednesday	- i need to make a plane out of garrett, i will be weaving town on wednesday.
- do you have any indian restaurants in the south in a different price range?	- do you have any indonesian restaurants in the south in a different prize range
- nope, same people.	- nope, same pupil.
- i'm looking for a college type attraction	- i'm looking for a knowledge-type action.
- yes, please book me a room for friday	- yes, please cook me a broom for friday
- yes, could you please email or fax me the fare amount, as well as the reference number?	- yes, could you please email or text me the fair amount, as well as thereference code?
- ois el shaddai a guest house or hotel?	iz let shadai a gest house or motel?
- great! i also need a train from mount pleasant to sabattus, please.	- great! i also need a strain from mount pleasant to suspicious, please
- yes, can you help me find a train that can take me from lovelock to abbot?	- yes, can you help me find a plane that can take me from love lock to rabbit?

Table 9: Examples of augmented utterances generated by injecting phoneme-level ASR-style errors. For each original utterance (left), the corresponding augmented version (right) includes substitutions that mimic realistic ASR recognition mistakes. Blue-colored phrases indicate changes in keywords that are used as slot values in DST, while orange-colored phrases represent overall ASR-style errors.

- Noisy audio environment: Incorporated authentic cafe and traffic noise from `https://freesound.org/` with a 10 to 20 Signal-to-Noise Ratio (SNR) and transcribed it using the Whisper large model.

- Paraphrased environment: When recording the audio, the text was paraphrased to resemble more natural, real-life spoken language(Soltau et al., 2022).

- High-acc ASR environment: Whisper-large model (1550M) is used for transcription. WER on LibriSpeech.test-clean is 0.027.

B.2 Comparison Methods

- AEDA (Karimi et al., 2021): We randomly inserted punctuation marks, effectively maintaining the original word order.
- EDA (Wei and Zou, 2019): We augmented data by applying edit-based technique that implements four rule-based modifications—synonym replacement, random insertion, swapping, and deletion.
- Back Translation (Sennrich et al., 2015): We translated original texts to error texts and then back to the original texts for generating syntactic variations during the process. We use English to German [3] and German to English[4] models as translator.
- TTS-ASR : We used Tacotron2 (Shen et al., 2018) for the TTS model to synthesize the audio and use Whisper-base (Radford et al., 2022) as an ASR model to simulate the ASR errors.
- ASR translation: We employed a sequence-to-sequence structure to translate clean text into ASR-errored text. Our training set comprised 300 hours of paired clean and ASR-errored text. We fine-tuned the model based on the T5-base architecture (Roberts et al., 2019), using the loss function defined in equation 3. The loss function is as follows:

$$L = -\sum_{i=1}^{I} \log P(e_i | g_i). \tag{3}$$

B.3 Training Details

In training models, we used T5-base (Roberts et al., 2019) as the backbone model and instructed the model to generate the B_t by given D_t in sequence to sequence manner, as in (Su et al., 2021) and the loss function is

$$L = -\sum_{t=1}^{T} \log P(B_t | \text{Inst}, D_t). \tag{4}$$

We set the learning rate as 4e-5 and used the AdamW (Loshchilov and Hutter, 2017) optimizer. One GeForce RTX 3090 is used for training and the batch size is 16. Trained until reaching the max patient, which is 3.

Method	Low-acc ASR		Noised Aud.		Paraphrased		High-acc ASR	
	JGA	N-acc	JGA	N-acc	JGA	N-acc	JGA	N-acc
Baseline	–	–	–	–	–	–	–	–
AEDA	ns	ns	ns	ns	ns	ns	ns	ns
EDA	*	*	**	**	ns	**	**	**
BT	**	**	**	**	*	**	**	*
TTS-ASR	ns	ns	ns	ns	ns	ns	ns	ns
ASR trans.	ns	*	ns	ns	ns	*	*	*
EPA (GPT-3.5)	***	**	***	***	**	**	*	**

Table 10: Statistical significance results compared to the Baseline using paired *t*-tests across three random seeds. Stars indicate significance levels: * for $p < 0.05$, ** for $p < 0.01$, *** for $p < 0.001$, and *ns* for non-significant differences.

C Further Experiments

C.1 Statistical Significance Analysis

To assess the reliability of our results, we conducted paired *t*-tests between each method and the Baseline to determine whether the observed performance improvements are statistically significant. We report 95% confidence intervals to reflect performance variability. Statistical significance is denoted using asterisks: $*$ for $p < 0.05$, $**$ for $p < 0.01$, and $***$ for $p < 0.001$.

As shown in Table 10, EPA (GPT-3.5) achieves statistically significant gains in nearly all evaluation settings, particularly under low-accuracy and noised ASR conditions. These results confirm that the improvements brought by our method are both consistent and statistically reliable.

C.2 Baseline Performance Comparison with Clean Text

For comparison, we report the baseline performance on an error-free, clean test dataset. Please note that DSTC11 (Soltau et al., 2022) does not provide a text script for the test dataset, so we are manually cleaning 50 dialogues to ensure they are error-free. In the experiment, the baseline model achieved a JGA score of 45.2 % and an N-ACC score of 86.5 % in an ASR error-free environment. Compared to the JGA, which is 34.8 %, and N-ACC, which is 52.07 %, in the ASR-errored environment (High-acc ASR model environment), this discrepancy highlights the significant impact of ASR errors on performance degradation.

C.3 Effect of Keyword Augmentation

To further validate the contribution of each component in our method, we conducted an ablation study separating Step 1 (overall error augmentation) and Step 2 (keyword-specific error augmentation), using GPT-3.5 as the generator. Both aug-

[3] facebook/wmt19-en-de

[4] facebook/wmt19-de-en

Method	Weak-ASR		Noised Audio		Strong ASR	
	JGA	N-acc	JGA	N-acc	JGA	N-acc
Baseline	29.88	45.76	29.70	46.77	34.87	52.07
+ Overall Error (Step 1)	31.31	50.67	31.13	52.29	35.40	55.80
+ Keyword Error (Step 2)	31.66	49.88	31.61	51.02	36.12	54.23
EPA (Step1+2)	**32.39**	**51.12**	**32.24**	**52.70**	**36.61**	**55.87**

Table 11: Ablation on overall and keyword-specific augmentation steps. Combining both (EPA) yields the highest robustness across ASR conditions.

Method	Low-acc ASR		Noised Aud.		Paraphrased	
	JGA	N-acc	JGA	N-acc	JGA	N-acc
Baseline	29.9	45.82	27.25	41.81	25.81	44.51
+ EPA	30.63	48.54	27.5	46.52	27.65	47.96

Table 12: Experiment with GPT-2 model as baseline.

Method	Low-acc ASR	Noised Audio	High-acc ASR
Baseline	56.29	60.50	60.02
+ OPT 6.7B	59.64	62.84	62.47
+ LLaMA 7B	58.29	60.53	60.46
+ GPT-3.5 (125B)	59.39	62.62	62.16

Table 13: NER results on the ASAPP/SLUE dataset under different ASR conditions. We reported the F1 score.

Method	Low-acc ASR	Noised Audio	High-acc ASR
Baseline	55.25	63.23	64.32
+ OPT 6.7B	56.97	64.94	66.10
+ LLaMA 7B	57.24	64.26	65.56
+ GPT-3.5 (125B)	58.78	66.16	67.49

Table 14: SLU results on the SLURP dataset under different ASR conditions. We reported F1 score.

mentation strategies individually improved performance over the baseline. Step 1 yielded consistent gains across all ASR settings, while Step 2 notably enhanced slot-value accuracy. When combined (EPA, ours), the model achieved the highest robustness in all evaluation conditions, confirming that the two strategies are complementary.

C.4 Experiments with a Different Baseline

In the main experiments, we use T5-base as the backbone model. To assess the generalizability of our approach, we additionally conduct experiments using a GPT-2 (Radford et al., 2019) model, as shown in Table 12. The results show a consistent trend with those of T5-base, demonstrating that our method is effective across different backbone architectures.

C.5 Generalizability to Other Tasks

To evaluate the generalizability of our approach beyond the DST domain, we extended our experiments to two additional spoken language understanding tasks: Named Entity Recognition (NER) and Spoken Language Understanding (SLU). We applied our EPA methodology under three ASR conditions—low-accuracy ASR, noised audio, and high-accuracy ASR—using the same experimental setup as in the DSTC11 evaluation. We used asap-p/slue dataset for NER task(Shon et al., 2022), and SLURP dataset (Bastianelli et al., 2020) for SLU task.

The results, shown in Table 13 and Table 14, demonstrate that our method consistently improves performance across all ASR conditions for both NER and SLU tasks. Notably, the gains are especially prominent under low-accuracy and noisy conditions, confirming that our approach is broadly applicable to other tasks.

C.6 Additional Fine-grained Metrics

Method	Precision	Recall	F1	Slot Accuracy
Baseline	50.5	50.1	50.3	92.8
+ TTS-ASR	49.5	49.6	49.5	93.4
+ ASR-Translation	51.8	51.7	51.7	93.1
+ EPA (GPT3.5)	55.2	54.8	55.0	93.7

Table 15: Fine-grained DST metrics under Low-accuracy ASR setting.

To supplement the main results focusing on JGA and N-Acc, we report additional fine-grained metrics—Precision, Recall, F1, and Slot Accuracy—under two ASR corruption settings: Low-accuracy ASR and Noised ASR. These metrics provide a more comprehensive view of model behavior in diverse error conditions in table 15 and 16.

C.7 Results with Different Random Seeds

Table 17 reports the results of our main experiments (Table 2) repeated with three different random seeds, demonstrating the consistency of the observed trends.

D Details of Quality Evaluation

D.1 About Metric

As described in Section 3.4, we use a phonetic similarity metric to evaluate pronunciation-level consistency between the original and augmented text. Specifically, we compute the normalized phoneme edit distance, which quantifies the minimal number of phoneme-level operations required to transform one utterance into another. A higher score indicates greater phonetic similarity. We used the

Method	Precision	Recall	F1	Slot Accuracy
Baseline	52.0	51.3	51.6	92.5
+ TTS-ASR	53.7	53.1	53.4	92.8
+ ASR-Translation	53.8	53.0	53.4	92.5
+ EPA (GPT3.5)	56.1	55.3	55.7	93.2

Table 16: Fine-grained DST metrics under Noised ASR setting.

Method	Low-acc ASR		Noised Aud.		Paraphrased		High-acc ASR	
	JGA	N-acc	JGA	N-acc	JGA	N-acc	JGA	N-acc
Baseline	30.05	46.48	29.80	47.49	29.08	48.85	34.73	52.31
AEDA	29.99	46.53	29.80	47.60	28.95	49.02	34.94	52.40
EDA	29.16	47.67	28.93	49.50	28.12	50.10	33.74	53.66
BT	31.54	49.21	31.43	51.25	29.90	51.60	36.25	54.85
TTS-ASR	30.09	46.16	30.32	47.74	29.26	49.31	35.28	51.95
ASR trans.	29.98	47.70	29.95	48.49	29.72	50.43	34.82	53.35
EPA (GPT3.5)	32.56	51.62	32.27	53.48	31.10	53.91	36.65	56.14
Baseline	29.82	45.63	29.75	46.20	28.55	48.75	34.83	51.44
AEDA	29.77	46.29	29.69	47.40	29.26	48.93	34.93	52.26
EDA	29.27	47.48	28.57	49.41	28.20	49.99	33.59	53.68
BT	31.66	49.16	31.05	50.81	29.88	51.75	36.19	54.73
TTS-ASR	29.75	45.99	29.80	46.87	29.10	48.12	34.97	51.49
ASR trans.	30.86	47.47	30.52	48.53	29.63	50.11	35.73	53.59
EPA (GPT3.5)	32.33	50.74	32.41	52.28	31.02	53.15	36.75	55.82
Baseline	29.77	45.18	29.56	46.61	29.12	48.76	35.05	52.47
AEDA	29.94	46.55	29.72	47.45	29.16	48.62	34.94	52.29
EDA	29.22	47.79	28.59	49.61	27.92	49.88	33.71	53.99
BT	31.86	49.13	31.31	50.87	29.91	51.85	36.38	54.85
TTS-ASR	29.98	46.86	29.84	47.51	28.88	49.20	34.97	52.66
ASR trans.	30.37	47.78	29.94	48.32	29.26	50.60	35.20	54.05
EPA (GPT3.5)	32.27	51.01	32.04	52.33	30.72	52.96	36.42	55.66

Table 17: Experiment result with different seeds.

`eng-to-ipa` library[5] for phoneme conversion in our implementation, as shown in the code snippet below.

```python
def phonetic_similarity(original_text,
     augmented_text):

    original_ipa  = to_phoneme(original_text)
    augmented_ipa = to_phoneme(augmented_text)

    edit_distance = nltk.edit_distance(
         original_ipa, augmented_ipa)

    # Normalize the edit distance
    max_length = max(len(original_ipa), len(
         augmented_ipa))
    normalized_distance = float(edit_distance) /
          float(max_length)

    # Convert to similarity score
    similarity_score = 1 - normalized_distance
    return similarity_score
```

D.2 Human Evaluation Details

To assess the plausibility of the generated ASR-style errors, we conducted a human evaluation involving two graduate students. Each participant was asked to rate whether a given sentence transformation could plausibly be attributed to an ASR error, using a 4-point Likert scale:

- **1 – Not at all**: The change is unlikely to be due to an ASR error. It appears to stem from other factors such as meaning alteration or stylistic variation.
- **2 – Unlikely**: The transformation is probably not caused by an ASR error.
- **3 – Somewhat likely**: The transformation may plausibly be caused by an ASR error.
- **4 – Very likely**: The transformation clearly appears to result from an ASR error.

Each sentence pair (original and transformed) was rated independently by both annotators. Inter-rater agreement and average scores are reported in Section 3.4. The distribution of Likert scores for each annotator is as follows : Annotator 1 assigned 5% of scores as 1, 7% as 2, 17% as 3, and 71% as 4. Annotator 2 assigned 3% of scores as 1, 24% as 2, 53% as 3, and 20% as 4.

[5] https://pypi.org/project/eng-to-ipa/