ConSLU: Constrained Decoding for Enhanced Spoken Language Understanding in Joint End-to-End Models

Dinh-Truong Do, Minh-Phuong Nguyen, Le-Minh Nguyen
Japan Advanced Institute of Science and Technology, Japan
{truongdo, phuongnm, nguyenml}@jaist.ac.jp

***Abstract*—Spoken language understanding (SLU) commonly employs cascading systems, integrating Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) modules. However, these systems often suffer from information loss, latency, and high costs, prompting significant research interest in end-to-end (E2E) SLU. Among E2E methods, joint models have emerged as particularly effective in terms of latency and accuracy. However, prior works on joint E2E SLU often represent the output logical form as a sequence of tokens, lacking a guarantee of producing a correct logical form. In this study, we enhance the joint E2E SLU approach by simplifying the output sequence and constraining the decoding process to focus on candidate tokens. Specifically, we categorize tokens in the logical form into label tokens and normal tokens, applying constrained candidates for each token type. Through experiments on the STOP dataset, our method outperforms previous works (by 1.44 exact match improvement compared to the baseline), achieving a 78% exact match score, demonstrating its effectiveness.**

***Index Terms*—Spoken language understanding, Constrained decoding, Joint models**

I. Introduction

The importance of spoken language understanding (SLU) in virtual assistant technologies, as evidenced by platforms like Siri, Alexa, and Google Assistant [1], [11]. Previous research has explored two primary methodologies in SLU: the cascade system, integrating ASR and NLU separately, and the end-to-end (E2E) approach, directly inferring semantic meaning from audio inputs. Given the prevalent deployment of virtual assistants on resource-constrained devices, E2E methodologies have gained popularity due to their efficiency in latency and information retention [12].

In E2E SLU system design, there are four main architectural approaches [5], [9]: (1) direct models, which learn semantic representations directly from input audio without utterance transcription; (2) joint models, which predict ASR transcripts and semantic representations simultaneously using a shared decoder; (3) multitask models, which employ separate decoders for ASR and NLU tasks while sharing an acoustic encoder; and (4) multistage models, resembling traditional cascading pipelines with distinct encoder-decoder structures for ASR and NLU, interconnected through gradient propagation. Among these, direct models show less promising results compared to others [9], indicating the crucial role of predicting ASR transcripts in deriving semantic forms due to the lack of information for logical label prediction. Among other architectures, joint E2E models offer the most promising approach as they achieve comparable results without increasing model size by adding extra components for ASR prediction, making them feasible for on-device settings. This paper adopts the joint E2E model, employing constrained decoding and simplifying output sequences to enhance spoken language understanding tasks.

Traditionally, most prior studies employing the joint E2E architecture have utilized autoregressive (AR) models built on the Transformer architecture [15]. In this setup, the decoding phase sequentially predicts tokens for both ASR and its logical form, typically starting with ASR tokens before proceeding to logical form tokens for better alignment. However, previous approaches treat the logical form as a sequence of tokens, necessitating consideration of the entire vocabulary at each step, which can lead to mislabeling and invalid logical form. For instance, when predicting a label token under the intent `IN:CREATE_ALARM`, it's unnecessary to consider the entire vocabulary; only candidate tokens relevant to the slots within the intent `IN:CREATE_ALARM`, such as `SL:DATE_TIME`, `SL:DURATION`, and others, need to be considered (Figure 1). Similarly, when predicting normal tokens, such as the 13th decoding step in Figure 1, the model do not need to consider the entire vocabulary; only tokens present in the ASR script are relevant. To address these issues, we propose a constrained methodology for training and inference in spoken language understanding. Drawing inspiration from prior work using grammar to constrain decoding steps [2], [3], we first extract grammar from the annotated training data to aid in predicting label tokens. This grammar informs the possible label tokens based on the parent node label. For example, if the current decoding output is `SL:DATE_TIME` and the parent node is `IN:CREATE_ALARM`, then, according to the grammar, the candidate label tokens should be possible slots for the intent `IN:CREATE_ALARM`: [`SL:DATE_TIME`, `SL:DURATION`, `SL:AMOUNT`, ...]. For normal tokens, we constrain the model to predict only those tokens present in the ASR script. During training, we mask all tokens not in the candidate set, allowing the model to focus on relevant tokens for better predictions. Additionally, unlike previous work [16], which employs multitask sequence spoken language understanding and represents output in a structured but challenging-to-predict JSON format,

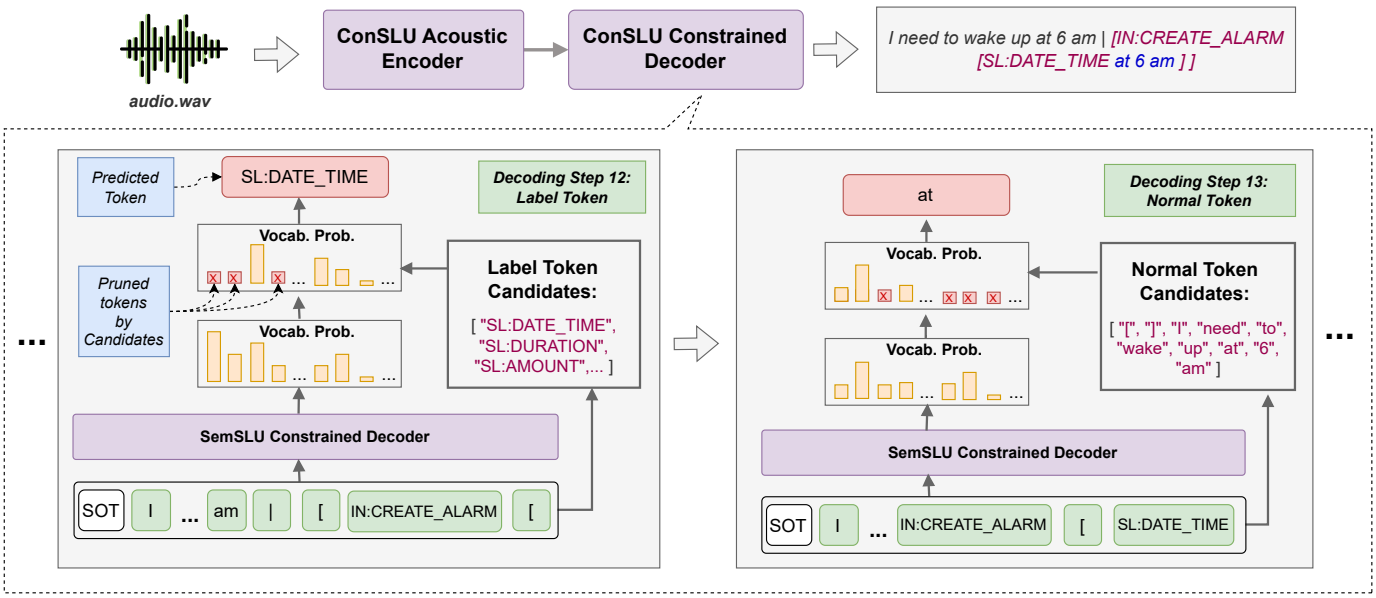

Fig. 1. Overview of our method

we hypothesize that not all multitask tasks are necessary, and the JSON format is ineffective as it is not consistent with the pre-trained task of pre-trained speech recognition models like Whisper [10]. Therefore, we simplify the output sequence to contain only two tasks: ASR and NLU, separated by a special token "|".

Our contributions in this study are threefold:

- We propose a constrained decoding approach to enhance the performance of joint E2E SLU systems.
- We demonstrate the effectiveness of simplifying the output sequence of logical form, which enhances performance compared to the structured yet challenging-to-predict JSON format.
- Through experimentation, we showcase the efficacy of our framework, surpassing current state-of-the-art joint E2E SLU methods with a 1.44 EM score improvement.

II. Related Works

The STOP dataset [14] is a well-known benchmark for assessing SLU system performance. This dataset adopts a hierarchical representation [4] for logical forms, enabling the inclusion of multiple intents within a single utterance. However, this structure also poses a challenge for models to accurately predict the correct logical form. To address this challenge, various approaches have been proposed. Kim et al. [8] introduced a method to enhance ASR error robustness by integrating audio and text representations based on estimated modality confidence of ASR hypotheses. Istaiteh et al. [7] employed a pre-trained HuBERT model [6] as an encoder alongside a transformer decoder with layer-drop and ensemble learning for decoding. Zhang et al. [17] proposed a two-stage approach for the SLU task. In the first stage, models based on encoder-decoder structures recognize speech utterances into text, while in the second stage, BERT with Conditional Random Field (CRF) and Byte Pair Encoding (BPE) are utilized for intent determination and slot filling in the SLU process. Wang et al. [16] introduced a sequence-level multitask learning paradigm, prioritizing tasks based on semantic complexity and concatenating their labels into a formatted JSON sequence for direct model learning. This approach facilitates smoother task transfer learning and enhances the main task's performance by leveraging auxiliary task predictions. In contrast, our method introduces a novel approach to constrain the decoding step of the spoken language understanding task. This approach ensures the model returns the correct logical form and enhances learning during training by focusing solely on the candidate set, while also improving inference by disregarding unpromising tokens.

III. Methodology

Figure 1 provides an overview of our methodology. Given an audio file, our model predicts both the ASR script and logical form directly. To accomplish this, the acoustic encoder first encodes the audio into audio embeddings. Using a constrained decoder, tokens are predicted step-by-step during decoding. Specifically, for each predicted token in the logical form, the probability distribution of the next tokens is calculated as usual. Subsequently, a candidate set is generated containing all possible tokens for that step. We then prune the vocabulary probability distribution, retaining only tokens present in the candidate set. The token with the highest probability from the pruned vocabulary distribution is selected as the next predicted token. Furthermore, the output sequence contains only the ASR script and logical form separated by a special token, facilitating fine-tuning steps to closely resemble the pre-trained task of models.

A. Output sequence simplified

Previous research [16] demonstrates the effectiveness of training models for spoken language understanding alongside other tasks, such as speaker gender classification, speaker native-ness classification, ASR, and domain classification, in improving SLU task performance. However, results from these studies also suggest that not all tasks are equally important for SLU. Therefore, we hypothesize that only the ASR task is necessary to enhance SLU performance. Based on this hypothesis, we simplified the output sequence text from multitask to include only two tasks: ASR and SLU. Additionally, instead of considering the sequence output as a well-structured but challenging-to-predict format like JSON, we simplified it by separating tasks with a special token "|", making the fine-tuning task more similar to the pre-trained task of pre-trained models. An example is illustrated in Figure 2.

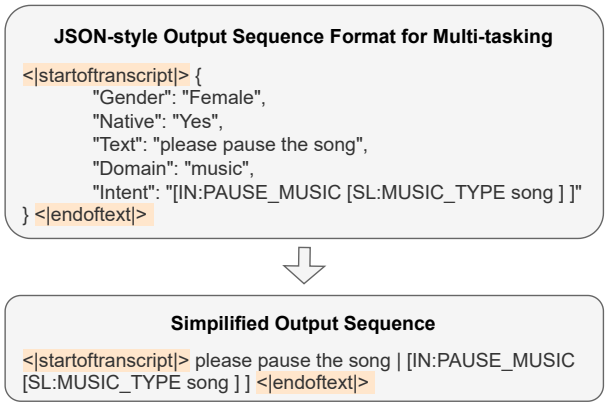

Fig. 2. An example of simplified output sequence

B. Constrained Decoding

The output sequence consists of two parts: the ASR script and the SLU logical form. For the ASR script, decoding proceeds normally using the audio embedding and the output of the previous decoding step. Upon completing the ASR script generation, we initiate SLU logical form generation, where we propose the use of constrained decoding. The problem with normal decoding when generating the logical form is that it does not guarantee a valid logical form and instead of considering the entire vocabulary, it is more efficient to consider only a small set of candidates at each decoding step. Therefore, we propose to use constrained decoding.

Specifically, there are two types of tokens in the hierarchical representation of the logical form: label tokens and normal tokens. We apply constrained candidates for each type of token.

- **Label token:** Following previous work [2], [3], we extract label grammar using training annotated data. This grammar serves as a candidate generator, providing a list of label candidates given the parent label of the current step. After obtaining the grammar, during the label token decoding step, we compute the vocabulary distribution probability for the next token as usual. We then generate the candidate set using this grammar. Subsequently, we prune all tokens in the vocabulary probability distribution that do not exist in the candidate set. The token with the highest probability in the pruned vocabulary probability distribution is selected as the predicted token (referred to as decoding step 12 in Figure 1).
- **Normal token:** For normal tokens, we constrain the next tokens to be tokens in the ASR script. This means that we extract all tokens in the predicted ASR script and set the candidate of the normal token to be in this set (referred to as decoding step 13 in Figure 1). Additionally, we add two special tokens "[", "]" to the candidates, which define the structure of the logical form.

Training: For mathematical modeling, given an audio speech $\mathcal{A}$, our model needs to predict the output $\mathcal{Y} = [y_1, ..., y_m, y_{m+1}, ..., y_n]$ where $\mathcal{Y}_{\mathcal{ASR}} = [y_1, ..., y_m]$ is the ASR script and $\mathcal{Y}_{\mathcal{SLU}} = [y_{m+1}, ..., y_n]$ is the output logical form. The loss function is defined as the negative log-likelihood of the true tokens in the output sequence, and it is defined as follows:

$$Loss_{ASR} = -\sum_{t=1}^{m} \log\left(\frac{e^{z_t}}{\sum_{v=1}^{\mathcal{V}} e^{z_{t,v}}}\right) \quad (1)$$

$$Loss_{SLU} = -\sum_{t=m+1}^{n} \log\left(\frac{e^{z_t}}{\sum_{c=1}^{\mathcal{C}} e^{z_{t,c}}}\right) \quad (2)$$

$$Loss = Loss_{ASR} + Loss_{SLU} \quad (3)$$

Here, $\mathcal{V}$ denotes the vocabulary, $\mathcal{C}$ represents the candidates for the current predicted logical form token, where $\mathcal{C} \subset \mathcal{V}$. In this context, z_t stands for the pre-softmax logit for the true token at time step t, and $z_{t,c}$ represents the pre-softmax logit for token v at time step t.

Inference: In the inference step, one challenge is determining whether the current token should be the label token or the normal token. We propose a simple solution based on the result of previous tokens in the logical form. If the last predicted token is "[", the next token should be the label token; otherwise, it is the normal token. Additionally, when predicting the normal token, the token should come from left to right of the ASR script. Therefore, if a token is selected as the normal token for the current step, the candidates for the next normal token step can only be to the right of this token in the ASR script. Inference continues until semantic form validation, where each open bracket token corresponds to a corresponding close bracket.

IV. Experiment

A. Dataset and Evaluation Metric

To assess the efficacy of our approach, we conducted experiments on the widely recognized spoken language understanding dataset STOP [14]. This dataset employs a hierarchical representation [4] to represent logical forms, allowing for the expression of utterances with multiple intents and nested slots.

TABLE I
MAIN RESULTS OF OUR METHODS WITH PREVIOUS WORKS ON STOP TEST SET.

Method	**Pre-trained Model**	**EM (%)**	**EM-Tree (%)**
wav2vec2.0 + Transformer Decoder [13]	wav2vec 2.0	68.70	82.78
HuBERT + Transformer Decoder [13]	HuBERT	69.23	82.87
Cascade system [13]	HuBERT+BART	72.36	82.78
WhiSLU [16]	Whisper-large	74.49	84.89
WhiSLU-SML [16]	Whisper-large	76.68	86.37
(Our methods)			
ConSLU-large	Whisper-large	77.38	87.37
ConSLU-large-v2	Whisper-large-v2	**78.12**	**87.63**

Covering a diverse range of SLU domains, STOP offers a rich variety of scenarios, comprising 82 distinct intents and 84 different slots. This diversity enables a comprehensive evaluation of our model's SLU capabilities. Consistent with prior research [16], we utilized the full-resource version of STOP, which encompasses 120k training, 33K development, and 75K test examples.

In line with previous studies [16], our primary evaluation metric was Exact Match (EM), where a score of 1 indicates a complete match between all predicted tokens and the ground truth sequence, and 0 otherwise. Additionally, we employed EM-tree as a secondary metric, which assesses whether the returned logical form correctly matches the semantic schema tree, albeit with potential inaccuracies in the predicted spans.

B. Experimental Settings

As our backbone models, we employed the pre-trained speech recognition Whisper model [10], utilizing various sizes including whisper-tiny, whisper-base, whisper-small, whisper-medium, and whisper-large. Initially, we ran each experimental hyperparameter setting on the whisper-base model to determine optimal values. Hyperparameters for other pre-trained models were kept consistent with those of the whisper-base. Our experimentation proceeded by first fine-tuning on the validation set to select the best hyperparameters, followed by evaluation on the test set to obtain final results. Specifically, we fine-tuned our models on the STOP dataset for {1, **3**, 5, 10}[1] epochs, using learning rates of {3e-05, **1e-05**, 5e-05}, 5000 steps of warmup, and a batch size of 16. Training on the whisper-base model with one GPU A100 took approximately 2 hours.

Baseline. To establish a strong baseline, we reproduced the WhiSLU model [16], adhering to the hyperparameters specified in the original papers.

C. Main Results

Table I shows the results of our methods with previous works on the test set of the STOP dataset. The results indicate that our best models outperform the top-performing method among joint E2E SLU models, WhiSLU [16]. Specifically, when comparing methods utilizing the same pre-trained model, whisper-large, our ConSLU approach exhibits superior performance to WhiSLU by 0.7 EM score. This highlights the effectiveness of simplifying the output sequence and employing constrained decoding in our method. Furthermore, utilizing the whisper-large-v2 model in our approach further enhances performance by 0.74 EM score. This underscores the scalability of our method to the efficacy of pre-trained language models; with improved pre-trained models, we can achieve superior performance when employing our method.

[1]Values in bold denote the best performance.

V. ANALYSIS

To gain deeper insights and analyze our proposed framework extensively, we conducted a comprehensive analysis.

A. Effect of model sizes

Table II presents the results obtained with different parameter sizes of Whisper. Larger LLM models consistently demonstrate better performance, with the whisper-large-v2 model achieving the best results, followed by the whisper-large and whisper-medium models. Notably, smaller models also prove effective. For instance, despite being only one-forty the size of whisper-large-v2, Whisper-tiny still achieves 90% of the larger model's performance. Similarly, Whisper-base, which is 20 times smaller than whisper-large-v2, attains 93% of its performance. This suggests that model size can be chosen based on specific computational constraints without significant performance trade-offs.

TABLE II
IMPACT OF LLM SIZE

Method	Parameters	**EM (%)**	**EM-Tree (%)**
whisper-tiny	39 M	70.04	82.14
whisper-base	74 M	72.51	83.89
whisper-small	244 M	75.02	85.52
whisper-medium	769 M	76.98	86.54
whisper-large	1550 M	77.38	87.37
whisper-large-v2	1550 M	**78.12**	**87.63**

B. Ablation Study

To evaluate the effect of each component in our framework, we compared the model's performance for each combination of component settings with that of the WhiSLU baseline model on the STOP dataset (Table III). Specifically, when utilizing the same pre-trained model whisper-base, our ConSLU method outperforms WhiSLU by 2.32 EM score, demonstrating the effectiveness of simplifying the output sequence and employing constrained decoding in our method. Using constraints led to improved performance on the STOP dataset (EM score 1.16 points higher than with the baseline model). Additionally, simplifying the sequence outputs improved the EM score compared with the baseline (0.98 points higher), highlighting the effectiveness of the two proposed points in our paper.

TABLE III
ABLATION STUDY

Method	EM-Tree (%)	EM (%)	Δ (EM)
SemSLU-base	**83.89**	**72.51**	_
- w/o constrained	82.79	71.35	-1.16
- w/o simplified	82.88	71.17	-1.34
Baseline	81.75	70.19	-2.32

C. Logical Form Validation

In this section, we compare the validity of the semantic representations generated by our approach and baseline approaches (Table IV). Unlike our approach, which generates perfectly valid trees when trained on the STOP dataset, WhiSLU struggles to achieve perfect validity. This demonstrates that the WhiSLU model requires larger model abilities to learn the structure of semantic representation to generate a valid logical form.

TABLE IV
THE ACCURACY OF RETURNING A VALID LOGICAL FORM OF BASELINE AND OUR METHOD CONSLU.

Method	Pre-trained Model	Logical Form Validation
WhiSLU	Whisper-base	99.59
ConSLU	Whisper-base	100.00

D. Case Study

Table V presents several examples outputted from our model and the baseline. In the first example, our model predicted the slot `SL:PERSON_REMINDED` after the intent `IN:DELETE_REMINDER`, whereas the baseline model predicted the slot `SL:ATTENDEE`. This difference occurred because the extracted grammar does not contain the constraint (`IN:DELETE_REMINDER => SL:ATTENDEE`). This demonstrates the effectiveness of using constrained decoding in our method. In the second example, the baseline correctly predicts the ASR script but fails to predict the logical form by predicting the wrong normal token "gabbage," while our model predicts the output correctly. This demonstrates the effectiveness of considering only tokens in the ASR script when predicting the normal token in the logical form. The final example is an instance where both our model and the baseline fail to return the correct output, predicting incorrect ASR scripts, leading to incorrect logical forms. This suggests that enhancing the ASR transcript can further improve the performance of our framework for future work.

CONCLUSION

In conclusion, our study delves into the realm of spoken language understanding (SLU), focusing on the joint E2E architecture, a pivotal methodology in virtual assistant technologies. We observe that while previous works predominantly adopt autoregressive (AR) models using Transformer architecture, the decoding phase of such models often encounters challenges in accurately predicting logical forms due to the consideration of the entire vocabulary at each step. To address this, we introduce a constrained decoding approach, leveraging grammar extraction to constrain the prediction space, thereby enhancing the accuracy of logical form predictions while reducing candidate size during token prediction. Additionally, we simplify the output sequence of logical forms to only contain essential information, departing from the intricate JSON structure typically used. Through experimentation, our framework demonstrates superior performance, surpassing current state-of-the-art joint E2E SLU methods by achieving a notable 1.44 EM score improvement. In essence, our contributions pave the way for more efficient and accurate spoken language understanding in virtual assistant technologies, offering a promising avenue for future research and development in this domain.

REFERENCES

[1] Jerome R Bellegarda. Spoken language understanding for natural interaction: The siri experience. *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice*, pages 3–14, 2013.

[2] Dinh-Truong Do, Minh-Phuong Nguyen, and Le-Minh Nguyen. Gram: Grammar-based refined-label representing mechanism in the hierarchical semantic parsing task. In *International Conference on Applications of Natural Language to Information Systems*, pages 339–351. Springer, 2023.

[3] Truong Do, Phuong Nguyen, and Minh Nguyen. Structsp: Efficient fine-tuning of task-oriented dialog system by using structure-aware boosting and grammar constraints. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10206–10220, 2023.

[4] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. Semantic parsing for task oriented dialog using hierarchical representations. *arXiv preprint arXiv:1810.07942*, 2018.

[5] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE, 2018.

[6] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

TABLE V
COMPARISON OF OUTPUTS OF BASELINE (WHISLU) AND OUR CONSLU MODEL ON THE STOP DATASET.

Type	Ouput
Ground-Truth	**ASR**: Please delete the movie reminder from the office group **Logical form:** [IN:DELETE_REMINDER [SL:TODO movie] [SL:PERSON_REMINDED office]]
Baseline	**ASR**: Please delete the movie reminder from the office group **Logical form:** [IN:DELETE_REMINDER [SL:TODO movie] [SL:ATTENDEE office]] ✗
ConSLU	**ASR**: Please delete the movie reminder from the office group **Logical form:** [IN:DELETE_REMINDER [SL:TODO movie] [SL:PERSON_REMINDED office]] ✓
Ground-Truth	**ASR**: Remind me to put garbage outside **Logical form:** [IN:CREATE_REMINDER [SL:PERSON_REMINDED me] [SL:TODO garbage outside]]
Baseline	**ASR**: Remind me to put garbage outside **Logical form:** [IN:CREATE_REMINDER [SL:PERSON_REMINDED me] [SL:TODO gabbage outside]] ✗
ConSLU	**ASR**: Remind me to put garbage outside **Logical form:** [IN:CREATE_REMINDER [SL:PERSON_REMINDED me] [SL:TODO garbage outside]] ✓
Ground-Truth	**ASR**: send happy birthday to jerilyn **Logical form:** [IN:SEND_MESSAGE [SL:CONTENT_EXACT happy birthday [SL:RECIPIENT jerilyn]]
Baseline	**ASR**: send happy birthday to jerrylenn **Logical form:** [IN:SEND_MESSAGE [SL:CONTENT_EXACT happy birthday [SL:RECIPIENT jerrylenn]] ✗
ConSLU	**ASR**: send happy birthday to jerryland **Logical form:** [IN:SEND_MESSAGE [SL:CONTENT_EXACT happy birthday [SL:RECIPIENT jerryland]] ✗

[7] Othman Istaiteh, Yasmeen Kussad, Yahya Daqour, Maria Habib, Mohammad Habash, and Dhananjaya Gowda. A transformer-based e2e slu model for improved semantic parsing. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE, 2023.

[8] Suyoun Kim, Akshat Shrivastava, Duc Le, Ju Lin, Ozlem Kalinli, and Michael L Seltzer. Modality confidence aware training for robust end-to-end spoken language understanding. *arXiv preprint arXiv:2307.12134*, 2023.

[9] Mohan Li and Rama Doddipatla. Non-autoregressive end-to-end approaches for joint automatic speech recognition and spoken language understanding. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 390–397. IEEE, 2023.

[10] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[11] Subendhu Rongali, Beiye Liu, Liwei Cai, Konstantine Arkoudas, Chengwei Su, and Wael Hamza. Exploring transfer learning for end-to-end spoken language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13754–13761, 2021.

[12] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE, 2018.

[13] Paden Tomasello, Akshat Shrivastava, Daniel Lazar, Po-Chun Hsu, Duc Le, Adithya Sagar, Ali Elkahky, Jade Copet, Wei-Ning Hsu, Yossi Adi, Robin Algayres, Tu Ahn Nguyen, Emmanuel Dupoux, Luke Zettlemoyer, and Abdelrahman Mohamed. Stop: A dataset for spoken task oriented semantic parsing. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 991–998, 2023.

[14] Paden Tomasello, Akshat Shrivastava, Daniel Lazar, Po-Chun Hsu, Duc Le, Adithya Sagar, Ali Elkahky, Jade Copet, Wei-Ning Hsu, Yossi Adi, et al. Stop: A dataset for spoken task oriented semantic parsing. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 991–998. IEEE, 2023.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] Minghan Wang, Yinglu Li, Jiaxin Guo, Xiaosong Qiao, Zongyao Li, Hengchao Shang, Daimeng Wei, Shimin Tao, Min Zhang, and Hao Yang. Whislu: End-to-end spoken language understanding with whisper. In *Proc. Interspeech*, volume 2023, pages 770–774, 2023.

[17] Gaosheng Zhang, Shilei Miao, Linghui Tang, and Peijia Qian. A two-stage system for spoken language understanding. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE, 2023.