TOP-DOWN ATTENTION IN END-TO-END SPOKEN LANGUAGE UNDERSTANDING

Yixin Chen[*]

Department of Statistics
University of California, Los Angeles (UCLA)

Weiyi Lu, Alejandro Mottini, Li Erran Li,
Jasha Droppo, Zheng Du, Belinda Zeng

Amazon Alexa
Seattle, USA

ABSTRACT

Spoken language understanding (SLU) is the task of inferring the semantics of spoken utterances. Traditionally, this has been achieved with a cascading combination of Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) modules that are optimized separately, which can lead to a suboptimal overall performance. More recently, End-to-End SLU (E2E SLU) was proposed to perform SLU directly from speech through a joint optimization of the modules, addressing some of the traditional SLU shortcomings. A key challenge of this approach is how to best integrate the feature learning of the ASR and NLU sub-tasks to maximize their performance. While it is known that in general, ASR models focus on low-level features, and NLU models need higher-level contextual information, ASR models can nonetheless also leverage top-down syntactic and semantic information to improve their recognition. Based on this insight, we propose Top-Down SLU (TD-SLU), a new transformer-based E2E SLU model that uses top-down attention and an attention gate to fuse high-level NLU features with low-level ASR features, which leads to a better optimization of both tasks. We have validated our model using the public FluentSpeech set, and a large custom dataset. Results show TD-SLU is able to outperform selected baselines both in terms of ASR and NLU quality metrics, and suggest that the added syntactic and semantic high-level information can improve the model's performance.

Index Terms— end-to-end SLU, top-down attention

1. INTRODUCTION

Spoken language understanding (SLU) systems aim to infer semantic concepts from spoken utterances. SLU systems are a key component of many modern applications, particularly of voice assistants such as Alexa, Siri and Google Assistant. As such, SLU has been extensively studied in recent years [1]. More concretely, given a spoken utterance, SLU will predict its domain, intent and slot values. Traditionally, this has been achieved with a cascading combination of an Automatic Speech Recognition (ASR) module in charge of transforming the speech signal into its textual representation, followed by a Natural Language Understanding (NLU) module that determines the semantics of the transcribed utterance. Although this approach has been highly successful, the fact that each module in the pipeline is designed and optimized separately has shortcoming that can lead to a suboptimal overall performance, including not directly optimizing for the final target metrics, and the fact that not all errors in ASR affect the NLU performance in the same way [2, 3]. To address these challenges, End-to-End SLU (E2E SLU) systems have recently gained popularity [4]. Under this approach, SLU is directly performed from speech features without necessarily requiring to produce an intermediate transcription of the speech. In addition, E2E SLU systems jointly optimize the modules, which in theory allows the components to leverage common features across tasks [2, 3]. However, an important challenge of this approach is how to best combine the feature learning of the ASR and NLU sub-tasks in order to maximize their performance. In general, ASR models focus on noisy low-level features (waveform, phonemes, etc.) [5], while NLU models require higher-level contextual information in order to perform their task correctly [6]. Nevertheless, it has been shown [7] that ASR models can also benefit from higher-level syntax and semantics to improve their recognition performance.

Following these insights, in this work we propose a new transformer-based E2E SLU model that learns how to best leverage lower- and higher-level features to jointly optimize both ASR and NLU tasks. The proposed approach, referred to as Top-Down SLU (TD-SLU), is based on Top-down Attention, a technique inspired from Cognitive Science [8, 9] and already successfully applied in the computer vision [10, 11] domain. More specifically, on top of the traditional transformer Sequence-to-Sequence (s2s) model [12] used in recent E2E SLU models [13, 14], we introduce an additional transformer encoder for the NLU task, which allows the model to utilize the entire sequence to gain semantic understanding by attention. In addition, following the multimodal literature, we combine the top-level NLU feature with the low-level acoustic features using an attention gate [15], which generates a shift in the low-level representation, adapting it according to the high-level information, thus improving the performance.

[*]Work performed while at Amazon

To validate our approach, we have compared the performance of our system against several baselines on the public Fluentspeech dataset, as well as on a large custom dataset. Results show that our model outperforms the other baselines in terms of both ASR and NLU metrics, and that the addition of high-level semantic information can benefit the ASR task.

2. RELATED WORK

Due to its advantages over traditional SLU systems, E2E SLU has been studied extensively in recent years [4, 2, 16]. In [2], the authors characterize four types of E2E SLU architectures: direct, joint, multi-task and multistage models. All four types follow the traditional encoder-decoder s2s architecture, with the difference lying in how the ASR and NLU sub-tasks are integrated. As such, subsequent E2E SLU models can be categorized as belonging to one of these classes. For example, [17] proposes a multi-stage model that uses a pretrained ASR module to fine-tune on SLU targets. Importantly, [17] also introduces the FluentSpeech dataset, which is routinely used in the E2E SLU literature. More recently, [13] proposes a direct model inspired by BERT that masks input acoustic features corresponding to particular output tokens, encouraging the model to leverage contextual information.

Overall, numerous E2E SLU works experiment with different architectural choices, but many miss exploring the relationship between the ASR and NLU tasks. To address this, we investigate ways of integrating the feature learning process of the tasks in a top-down manner, better capturing their intrinsic characteristics. We base our work on the cognitive science literature [8, 9], which has shown that both top-down and bottom-up information is critical for robust and accurate perception in humans. This approach has been extensively used in the visual domain to enable deeper understanding through fine-grained analysis and multi-step reasoning, and applied to diverse use-cases including image captioning [10] and visual question answering [11]. Moreover, [18] uses a RNN architecture in which bottom-up and top-down signals are dynamically combined using attention, and shows its validity in language modeling tasks. Closer to our work, the ASR task can be seen as taking in speech data, and abstracting acoustic features to perform the transcription, while the NLU module extracts semantic information from contextual information. In this sense, NLU is naturally a top-level task compared to ASR. The optimal combination of bottom-up and top-down information between these tasks remains an open question.

3. MODEL ARCHITECTURE

Following the characterization of [2], our model falls into the multi-stage category, the most similar to the conventional cascading SLU approach. This choice allows us to design the ASR and NLU modules separately, and later concentrate on leveraging the relationship between the tasks using top-down attention. Next, we introduce the encoder and decoder modules for the ASR and NLU tasks, followed by a description of the top-down attention mechanism that links them together.

3.1. ASR and NLU Modules

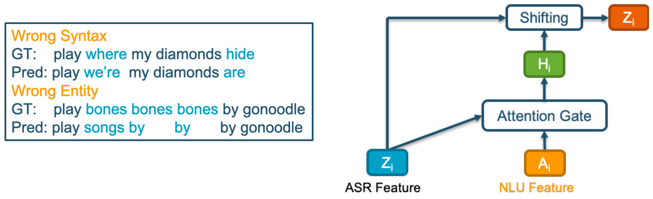
Our ASR module's architecture is similar to other recent [19, 18] transformer-based models. It follows the traditional encoder-decoder framework that first processes the input speech sequence, represented using its spectrogram, and decodes the target text sequence. More precisely, the input spectrogram is first fed into convolution and max pooling layers to extract features, which are fed into a transformer encoder. This encoder converts the input sequence into a shorter sequence of hidden states to be consumed by the transformer decoder. We use a standard embedding matrix to represent the target sequence's tokens. During the decoding process, a fixed BOS special token is used to start decoding. Then, the transformer decoder consumes the current token embedding, and performs a multi-head multi-layer attention over the encoder's hidden states to generate the decoder hidden state. These decoder hidden states have two uses. First, they are passed through a generator layer (the ASR2 block as depicted in Figure 2), which assigns a probability mass to each token in the target vocabulary, representing the probability of that token being generated next. Finally, when the decoding process ends, the decoded hidden vectors will be fed into the transformer encoder specifically designed for NLU.

For the NLU task, we propose to use a separate transformer encoder to better utilize the hidden features of the low-level ASR task. This choice is based on the extensive use of this non-recurrent neural architecture in most current text applications [12]. The superior performance of the transformer model is largely credited to a Multi-head Self-Attention module. Using this module, each element of a sequence is attended by conditioning on all the other elements. This capability makes transformers well suited for NLU, allowing it to look at the sentence as a whole to understand the semantic and syntax information of the utterance.

3.2. Top-down Attention Gate

While ASR errors can be due to many issues (background noise, multiple speakers, etc.), errors made due to rare words or confusion of phonetically similar words can be attenuated using higher-level semantics. For example, in Figure 1a, the prediction "play we're my diamonds are" is incorrect in terms of syntax and grammar. In the second example, knowing that "bones bones bones" actually belongs to the same NLU slot type ("SONGNAME") could help avoid the error. To address these issues, we propose to leverage high-level features from the NLU task to help the learning of the ASR task.

Inspired by the work of [15] on multimodal sentiment analysis, we use an attention gate to fuse the high-level feature from the NLU encoder with the low-level feature of the ASR decoder. Normally, for the ASR task, the latent space representation of individual tokens is directly conditioned on


Wrong Syntax
GT: play where my diamonds hide
Pred: play we're my diamonds are
Wrong Entity
GT: play bones bones bones by gonoodle
Pred: play songs by by by gonoodle

(a) ASR errors where syntax and entity information can help.

(b) Architecture of the top-down attention gate.

Fig. 1: Top-down attention for correcting ASR errors.

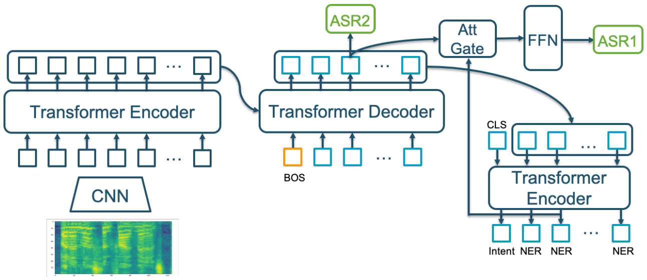


Fig. 2: Architecture of proposed end-to-end SLU model (TD-SLU) with top-down attention.

the input speech features. However, leveraging higher-level semantics can have an impact on the interpretation of each token, and therefore on their position in this space. By combining both representations with a gate, we can better extract what is helpful in the high-level features, modeling the interaction between both levels (with the attention score) to later shift the speech feature in the original latent space. A simpler alternative would be to concatenate/add both sets of features, but this would have a limited effect in adapting the ASR representation to consider the lexical information in the NLU features, or could even destroy the original representation.

More formally, let (A_i, Z_i) denote the features from NLU and ASR for the i_{th} word in a sequence. We first concatenate A_i with Z_i and compute the gating vector g_i. The gating vectors highlight the relevant part in the NLU features A_i. We then compute the displacement vector H_i by multiplying the NLU features with the gate vector. Finally, we sum the original ASR features with the corresponding transformed NLU features after the attention gate:

$$\begin{aligned} g_i &= R(W_g[Z_i, A_i] + b_g) \\ H_i &= g_i(W_h A_i) + b_h,\ \bar{Z}_i = Z_i + H_i \end{aligned} \tag{1}$$

where W_h, W_g, b_g and b_h are weights, and $R(*)$ is a non-linear activation function (our implementation uses sigmoid activation). $\bar{Z}_i$ is used to compute the the probability distribution over the target vocabulary using another generator layer (ASR1 block in Figure 2). It should be noted that the use of this top-down attention gate is not restricted to our model, and could also be used in other E2E SLU models such as [19, 18].

Figure 2 summarizes our architecture. Most notably, our model has two generation blocks to produce ASR prediction: ASR1 for features with top-down attention, and ASR2 for features directly produced by the speech transformer decoder. The reason behind this choice is that, during inference time, we perform beam search at ASR2 to get the 1-best output Z_i, and feed it into the NLU encoder to obtain the NLU features A_i. Then we take Z_i and A_i to compute $\bar{Z}_i$, which is used to generate the final output at ASR1. During training, instead of beam search, we use teacher forcing at ASR2 to obtain Z_i. We have observed that this design choice improves our performance considerably. In Section 4 we will discuss the differences between both ASR block outputs.

4. EXPERIMENTAL VALIDATION

We have conducted experiments on two datasets. The first one is FluentSpeech [17], which contains English utterances that one might use for controlling a virtual assistant. This data focuses on three types of SLU entities: {action, object, location}. There are approximately 23K training, 3K validation, and 4K test samples in this dataset. The vocabulary size is 100. For comparison, we follow [17] and compare the SLU accuracy (match of entire predicted hypothesis to gold hypothesis) between models. The second dataset is an internal dataset of de-identified utterances from a real-world voice assistant. In total, it is comprised of 2400 hours of speech for training, 270 hours for validation, and 100 hours for testing, and contains 23 intents and 95 slots from the music domain. The vocabulary size is 74k. Each utterance in this set contains the speech, text transcript, intent and the NER labels. Here, we evaluate the models using metrics commonly used to evaluate voice assistants' ASR and NLU engines, namely Word Error Rate (WER) and Semantic Error Rate (SemER) [3], a metric designed to reward partially correct NLU hypothesis.

Hyper-parameters were optimized using random search to optimize validation set performance. We use 80-dim log-filter bank features to represent the speech signals. For speech embeddings, we use a 2-layer 2D CNN with 256 final units. Both speech and NLU transformer encoders have 12 layers, 4 heads, 256 units, and 2048 hidden units. The speech decoder consists of a transformer decoder of 6 layers, 4 heads, 256 units, and 2048 hidden units. Our speech embedder, which includes the CNN layers and the speech transformer encoder, is pretrained on LibriSpeech [20]. The entire model is trained with Noam Learning rate schedule with 4000 warm-up steps and an Adam optimizer with learning rate of 1. We use the sequence cross entropy loss for the ASR1, ASR2 and NER tasks, and cross entropy for the Intention Classification task.

As baselines, we first consider a typical E2E SLU seq2seq model with a transformer encoder-decoder (referred to as E2E SLU), where the encoder is pretrained on LibriSpeech [20]. In addition, we perform a simple ablation study and propose two other baselines. The first one (referred to as TD-SLU-noAtt) uses an architecture similar to Figure 2, but without any connection between the ASR and NLU tasks. As such, the NLU encoder of Figure 2 is not connected back to the ASR decoder, and the ASR1 task is removed. The second one (TD-SLU-Concat-Att) follows the architecture of Figure 2, but replaces the attention gate with a simple concatenation of the ASR and NLU features. Finally, for the experiments conducted on FluentSpeech, we also compare with the best reported model in [17], which uses a RNN-based network (RNN-based).

4.1. Results

FluentSpeech Dataset: Table 1 shows the results on the FluentSpeech dataset. We can see that our proposed model (TD-SLU) achieves a near perfect SLU accuracy of 99.57, surpassing the best performing method in [17] (RNN-based model), as well as the E2E SLU baseline. This suggests that by connecting the ASR and NLU module, we can achieve a performance boost over E2E SLU systems that do not explore such connection between the tasks. TD-SLU also outperforms the models in the ablation study, particularly the variant without attention, which actually performs worse than E2E SLU.

Method	SLU Acc
E2E SLU	99.47
RNN-based [17]	98.8
TD-SLU-noAtt	99.45
TD-SLU-Concat-Att	99.55
TD-SLU	**99.57**

Table 1: Results on Fluentspeech dataset

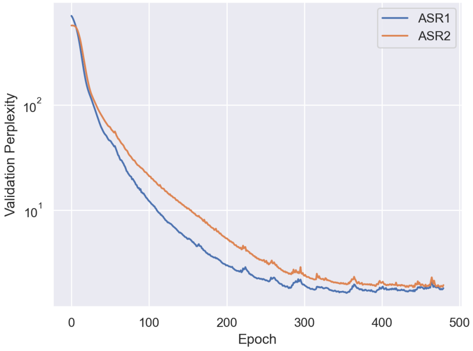


Fig. 3: Validation perplexity of ASR blocks during training.

Moreover, Figure 3 shows the validation perplexity of both ASR output blocks (ASR1 with top-down attention, and ASR2 with only the features produced by the transformer decoder). It can be observed that during the very early stage of training, ASR1 is outperformed by ASR2. We suspect the reason behind this is that when training just begins, the transformer encoder in the NLU module is not able to extract useful features and thus, we are only injecting non-sensical information to the ASR1 block through top-down attention. This harms the performance at first. However, as training proceeds, the NLU module learns to extract better features, which in turn benefits the ASR1 block. As a result, we can see that ASR1 consistently produces lower perplexity than does ASR2 after a few dozens of epochs. This furthers validates our hypothesis that the ASR task can benefit from higher-level features from the NLU task.

Method	WER	SemER
E2E SLU	1	1
TD-SLU-noAtt	1.11	1.05
TD-SLU-Concat-Att	0.99	0.96
TD-SLU	**0.94**	**0.93**

Table 2: Results on custom dataset, relative to E2E SLU

Internal Data: The resutls on the custom dataset are presented in Table 2, and concur with FluentSpeech results. When comparing the performance of E2E SLU and TD-SLU, we observe improvements both in terms of WER and SemER, which once again validates the benefit of top-down attention for modeling the interaction between the ASR and the NLU tasks. In addition, there are also some important findings from the ablation study. First, TD-SLU-noAtt actually produces the worst results. Recall that this model uses an additional encoder for the NLU task similar to TD-SLU, but without the top-down attention connection. As such, results suggest that the added complexity of the NLU encoder alone does not bring in extra benefit, and the improvement of TD-SLU mainly comes from the interaction between the ASR and the NLU tasks through top-down attention. Meanwhile, TD-SLU-Concat-Att, which uses top-down attention but only through a simple concatenation of features, is able to outperform E2E SLU, but by a slight margin. This suggests that the gated attention used in TD-SLU, with its 17% and 12% reduction in WER and SemER over TD-SLU-noAtt, is a more effective way of fusing the top- and bottom-level features. We believe similar gains could be observed by integrating this mechanism in other E2E SLU models.

5. CONCLUSION

We have presented TD-SLU, a novel transformer-based E2E SLU model that, unlike other current models, uses an attention gate to fuse high-level NLU features with low-level ASR features in a top-down manner. TD-SLU also includes a separate transformer encoder for the NLU task to leverage the decoded low-level ASR features. These architectural decisions are based on the insight that ASR models can benefit from high-level semantic information, and can generalize to other existing E2E SLU models. We have tested TD-SLU and selected baselines on the public Fluentspeech set and on an internal dataset. Results show that TD-SLU outperforms the baselines, and that the added semantic high-level information can benefit the model's performance. In the future, we wil explore the use of bottom-up attention, which has shown promises in the vision community.

6. REFERENCES

[1] Gokhan Tur and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[2] 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 *IEEE SLT Workshop*, 2018.

[3] Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rastrow, “Speech to semantics: Improve asr and nlu jointly via all-neural interfaces,” in *INTERSPEECH*, 2020.

[4] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio, “Towards end-to-end spoken language understanding,” in *ICASSP*, 2018.

[5] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., “English conversational telephone speech recognition by humans and machines,” in *INTERSPEECH*, 2017.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019.

[7] Benjamin E Lambert, Scott E Fahlman, Bhiksha Raj, Roni Rosenfeld, and Candy Sidner, *A Knowledge-Based Architecture for using Semantics in Automatic Speech Recognition*, Ph.D. thesis, Carnegie Mellon University, 2009.

[8] Ronald A Kinchla and Jeremy M Wolfe, “The order of visual processing: “top-down,”“bottom-up,” or “middle-out”,” *Perception & psychophysics*, vol. 25, no. 3, pp. 225–231, 1979.

[9] Karsten Rauss and Gilles Pourtois, “What is bottom-up and what is top-down in predictive coding?,” *Frontiers in psychology*, vol. 4, pp. 276, 2013.

[10] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *CVPR*, 2017.

[11] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *CVPR*, 2018.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *NIPS*, 2017.

[13] Chengyi Wang, Yu Wu, Yujiao Du, Jinyu Li, Shujie Liu, Liang Lu, Shuo Ren, Guoli Ye, Sheng Zhao, and Ming Zhou, “Semantic mask for transformer based end-to-end speech recognition,” in *INTERSPEECH*, 2020.

[14] Martin Radfar, Athanasios Mouchtaris, and Siegfried Kunzmann, “End-to-end neural transformer based spoken language understanding,” in *INTERSPEECH*, 2020.

[15] Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque, “Integrating multimodal information in large pretrained transformers,” in *ACL*, 2020.

[16] Hong-Kwang J Kuo, Zoltán Tüske, Samuel Thomas, Yinghui Huang, Kartik Audhkhasi, Brian Kingsbury, Gakuto Kurata, Zvi Kons, Ron Hoory, and Luis Lastras, “End-to-end spoken language understanding without full transcripts,” in *INTERSPEECH*, 2020.

[17] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, “Speech model pre-training for end-to-end spoken language understanding,” in *INTERSPEECH*, 2019.

[18] Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio, “Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules,” in *ICML*, 2020.

[19] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., “A comparative study on transformer vs rnn in speech applications,” in *IEEE ASRU Workshop*, 2019.

[20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015.