INCREMENTAL ONLINE SPOKEN LANGUAGE UNDERSTANDING

Prashanth Gurunath Shivakumar, Naveen Kumar, Panayiotis Georgiou, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory
University of Southern California, Los Angeles, California, U.S.A

pgurunat@usc.edu, komathnk@usc.edu, georgiou@sipi.usc.edu, shri@sipi.usc.edu

ABSTRACT

Spoken Language Understanding (SLU) typically comprises of an automatic speech recognition (ASR) followed by a natural language understanding (NLU) module. The two modules process signals in a blocking sequential fashion, i.e., the NLU often has to wait for the ASR to finish processing on an utterance basis, potentially leading to high latencies that render the spoken interaction less natural. In this paper, we propose recurrent neural network (RNN) based incremental processing towards the SLU task of intent detection. The proposed methodology offers lower latencies than a typical SLU system, without any significant reduction in system accuracy. We introduce and analyze different recurrent neural network architectures for incremental and online processing of the ASR transcripts and compare it to the existing offline systems. A lexical End-of-Sentence (EOS) detector is proposed for segmenting the stream of transcript into sentences for intent classification. Intent detection experiments are conducted on benchmark ATIS dataset modified to emulate a continuous incremental stream of words with no utterance demarcation. We also analyze the prospects of early intent detection, before EOS, with our proposed system.

Index Terms— Incremental Processing, Online Processing, Spoken Language Understanding, Intent Detection, Recurrent Neural Network

1. INTRODUCTION

With the proliferation of novel interactive technology applications across domains ranging from entertainment to health, the use of spoken language for enabling and supporting natural communication is becoming ever more important. Today, SLU finds applications in voice assistants, robot interactions, virtual agents, virtual & augmented reality applications as well as in mediating human interactions such as meetings. While the rapid development in SLU techniques has led to a revolution in this field, a lot still remains to be bridged in terms of improving the “naturalness” of these interactions. For example, achieving low latencies remains crucial to achieve a sense of “naturalness” during conversations. Higher latencies often result in turn-based disruptive conversations which creates the impression of a transactional interaction.1[2]

Typical gaps between human-human dyadic turns are of the order of 200 ms.1[2,3,4] In contrast, most ASRs rely on Inters Pausal Units (IPU) that are upwards of 500 ms to reliably detect the “end of utterance”. During interactions, this latency is often perceived as computational delay due to speech recognition, whereas in reality this delay can be avoided by use of incremental processing architectures.

Previous works such as 5[6,7] have investigated the stability of results when using incremental hypotheses from a speech recognition system. While these incremental hypotheses, often referred to as partials, can be generated with low latency, their semantic stability for downstream NLU tasks remains challenging. Earlier attempts to alleviate this issue has tried to use a trained confidence score 8[9,10] to assess stability of the prediction. Some researchers have also relied instead on auxiliary methods to predict turn-taking behavior in an agent.11[12,13,14]

Most of the research efforts in SLU in the NLP community assume offline ASR processing, i.e., (i) ideal, perfect utterance boundaries, and (ii) error-less transcriptions void of any speech recognition errors 15[16,17,18,19,20,21,22]. Leading NLU systems are based on RNN 15[16,17,18], convolutional neural network (CNN) 22 and sequence-to-sequence architectures 20 with attention modeling 20,21. Joint modeling of SLU tasks like intent detection, language modeling (LM), slot-filling, and named entity detection are found to be beneficial 15,16,17,18,19,20,21. Character level features have also been proposed to achieve state-of-the-art performance in benchmark tasks 15,22. Whereas, research efforts in the speech community assume offline ASR and offline NLU processing, i.e., (i) ideal, perfect utterance boundaries, and (ii) ASR errors in transcriptions 23,24,25,26. To handle ASR errors, joint SLU-ASR adaptation 27 and joint learning of SLU and ASR error correction 25,26 have been studied. Better feature representations involving acoustic information are beneficial in handling ASR errors for SLU 23,24. Although, there have been a few attempts at end-to-end SLU directly from speech signals, the performance are not quite up-to the standards achieved by the traditional approaches involving ASR and NLU 23.

As discussed earlier, few research efforts have tried to incorporate incremental processing on ASR 5[6,7]. Even fewer efforts have been made in the context of ASR incremental processing for SLU. The authors in 19 proposed a joint online SLU and LM system using RNN. Although, their proposed model is capable of outputting the intent class posterior for each time-step, the posteriors were not used for intent prediction itself, but were fed back to the hidden state of the RNN. The authors only consider the intent output at the last time-step of the input sequence for intent classification. Moreover, the evaluations were made in an offline fashion assuming each input sequence equals a single sentence and assuming the sentence boundaries were known a-priori.

To the best of our knowledge, there has been no prior work dealing with incremental online SLU employing RNN with evaluations conducted in a truly online sense. In this work, we setup the online incremental SLU processing along with (i) detection of utterance boundaries, and (ii) assumption of error-less transcriptions. The proposed system is capable of recognizing intents on an arbitrarily long sequence of words with no sentence or utterance demarcations.
2. NEED FOR INCREMENTAL ONLINE PROCESSING

To motivate the need for incremental online processing, in a real-life, real-time processing system, we present two different scenarios:

**Scenario 1: Endpoint-based Processing:** In a real-time application scenario, the ASR receives a stream of continuous speech signal and outputs the corresponding transcriptions in real-time. Due to the computational complexity and memory constraints, most ASRs typically operate by chunking and processing the speech in segments. This process is often referred to as end-pointing, and is usually determined based on different heuristics related to duration of IPUs, with the goal to minimize disruption during speech. Finally, the ASR outputs the transcript corresponding to each speech segment. In this scenario, the ASR is tuned for real-time application, by varying the parameters for end-pointing, often in a heuristic way. As a result, any application operating on the output of the ASR needs to wait at least until end-pointing, which gives rise to a fundamental bottleneck in latency.

**Scenario 2: Incremental Processing:** Alternatively, during ASR decoding, intermediate querying of ASR output transcript is possible. This involves computation of the best path over intermediate, incomplete decoded lattices (see example in Figure 2). Although, there is a possibility of the best path deviating between the complete and incomplete lattice decoding, the deviation is expected to be minimum in robust ASR systems. Moreover, the prospects of using incremental outputs of the ASR is attractive. In this scenario, the downstream application has no constraints of waiting until end-pointing and is free to process the ASR transcripts in an incremental manner. This also allows for online processing for downstream application in addition to the ASR itself for optimal latency considerations.

2.1. Incremental Processing for SLU Tasks

In the context of SLU, under **Scenario 1**, the NLU module is run in an offline fashion, processing an utterance at each end-point of the ASR. The timeline is illustrated in Figure 1A & 1B. It is evident from the time-line that offline NLU processing has higher latency implications. Moreover, here, the end-pointing algorithm itself, has a bearing on the performance of the NLU system, since end-pointing defines the utterance boundaries fed to the NLU. There have been several research efforts in predicting optimal end-point for an ASR [20, 21]. Thus, the NLU has to deal with the errors from: (i) ASR, (ii) end-point detection, and (iii) ASR errors due to sub-optimal end-pointing. Note, sub-optimal end-pointing especially false alarms can result in errors during recognition itself [22]. The aggregated errors often lead to degradation in the overall performance of SLU.

In this paper, we propose an SLU system under **Scenario 2**, where the NLU module can also be run in an online fashion, in parallel with the ASR. Additionally, we also propose incremental processing independent of ASR end-pointing and a lexical end-of-sentence (EOS) detection module for utterance boundaries, operating on the ASR output. This allows for lenient end-pointing schemes (emphasis on lower false positives) since end-pointing no longer defines latencies. This comes with the advantage that the NLU has to deal with errors from only the ASR phase. However, note that the EOS module might still introduce errors into the system, due to improper segmentation. The time-line of the incremental processing system is illustrated in the Figure 1D. It is apparent that there are significant latency advantages associated with the proposed system.

2.2. Implications on Neural Network Architectures

The online incremental nature of the NLU module imposes certain design constraints on the architecture of the recurrent neural networks. One of the fundamental restrictions due to the online nature of the problem is the use of only uni-directional LSTM. This is because, we don’t have access to the future time-steps for the backward step as in the case of bi-directional LSTM. Second, the incremental processing restricts the use of context to one for each time-step.

3. PROPOSED TECHNIQUES

3.1. Baseline Offline RNN

The baseline system consists of a vanilla RNN LSTM architecture which consumes a sequence of words and outputs a single decision similar to most of the works [15, 16, 17, 18, 19, 20, 21] with the exception that the LSTM is uni-directional as per Section 2.2. The network is referred to as offline, since the input needs to be segmented such that each utterance has a single intent label during training. However to assess its performance for the online task, during evaluation, we derive the output per each time step by sharing the output layer over all the time-steps. The latency implication of the system is illustrated in Figure 1A.

3.2. Online RNN Classification

The online version of the system is similar architecture wise to the baseline offline model with the exception that each input time-step has a corresponding output. The network is referred to as online, since it can process arbitrary length sequences with sequences of multiple intent labels both during training and testing. The system is trained with input sequences comprising multiple sentences/utterances which possibly map to a sequence of multiple different intents. During training, since each utterance has a single lat-
bel, we mask the loss function to compute the loss only over the utterance boundaries. The loss function is given by:

$$\text{Loss} = - \sum_{t=1}^{T} I_{EOS} \sum_{c=1}^{C} y_{o,c} \log p(y_{o,c})$$  \hspace{1cm} (1)$$

where \(t\) is the time-step, \(T\) is the sequence length, \(I_{EOS}\) is the indicator function which is 1 for oracle EOS and 0 otherwise, \(c\) is the intent class, \(C\) is the total number of intent classes, \(y_{o,c}\) is the indicator function which is 1 if the observation \(o\) belongs to class \(c\) and 0 otherwise, and \(p(y_{o,c})\) is the softmax probability prediction for observation \(o\) and class \(c\). The latency of the online system is pictured in Figure 1D.

### 3.3. End-of-Sentence Detection

Both the baseline Offline and Online RNN system do not have a sense of utterance boundaries during the evaluation phase. Thus, as described in Section 2.1, we train an EOS classifier independently. The architecture of the EOS detection system is similar to the online RNN Classification system with two exceptions: (i) it uses a sigmoid activation at the output, and (ii) binary cross-entropy to classify EOS. In conjunction with the EOS system the latency of baseline offline system and online system is pictured in Figure 1B.

### 3.4. Online Multi-task Learning

Additionally, we propose to model both the tasks i.e., intent detection and EOS detection jointly in a multi-task learning framework. In this framework, both the tasks share the embedding layer, but have task specific LSTM and time-distributed linear output layer (see Figure 2) with exception of feedback. The loss optimized is given by:

$$\text{Loss} = - \sum_{t=1}^{T} I_{EOS} \sum_{c=1}^{C} y_{o,c} \log p(y_{o,c})$$

$$+ y_{e} \log(p_e) + (1 - y_{e}) \log(1 - p_e)$$  \hspace{1cm} (2)$$

where \(y_e\) is the oracle EOS label, \(p_e\) is the predicted output of the network, the rest of the parameters comply with equation 1. The proposed system has two advantages: (i) the latency is further reduced since both tasks are modeled together, and (ii) joint learning of two tasks can benefit each of the tasks as demonstrated in [15] [16] [17] [18] [19] [20] [21].

### 3.5. Online Multi-task with EOS Feedback

Further, within the multi-task learning framework, we experiment with feeding back the EOS output back to the intent detection LSTM. The embedding layer is shared between the two tasks, with task specific LSTM layers and time-distributed linear output layers. The predicted EOS output is concatenated along with the input features from the embedding layer and fed to the intent LSTM (see Figure 2). The loss function is identical to the multi-task learning in equation 2. With this proposed framework, we believe that explicitly feeding the EOS markers to the intent detection system could provide performance benefits. The latency is identical to the multi-task learning framework described in section 3.4.

### 4. DATA & EXPERIMENTAL SETUP

#### 4.1. Data

We employ the ATIS (Airline Travel Information Systems) benchmark dataset [32] for performing our experiments on intent detection. The dataset consists speak recordings of humans speaking to an automated airline travel inquiry systems. The speech recordings are accompanied by manual transcriptions of the spoken queries with annotated intent labels which are used in this study. The data consists of 17 unique intent categories. Our setup is identical to [24] [17] [21].

### 4.2. Experimental Setup

To simulate online continuous stream of transcriptions, random number of samples from manual transcripts of the ATIS dataset were stitched together without exceeding a maximum number of utterances limit (see example in Figure 2). This results in each sample containing multiple utterances with sequence of multiple intent labels with no demarcation. We create multiple copies of the dataset by varying maximum number of utterances limit from 1 to 10 for analysis purposes. Note, the data contains exactly the same information and is consistent with previous studies involving ATIS dataset [24] [17] [21] to facilitate direct comparisons.

The RNN-LSTM models are trained on the samples from the training set and the development set is used for hyper-parameter tuning. Finally, the model with the best performance on the development set is chosen and evaluated on the unseen, held out test set. A single layer LSTM model is adopted with the embedding layer dimension set to 556 taking recommendations from [24]. The hidden dimension of LSTM was tuned over 32, 64, 128, 256 units. The dropout is varied over 0.1, 0.15, 0.20, 0.25, 0.3. The batch size is set to 1 with each sequence of utterances viewed as a single sample. The learning rate of 0.001 is used along with the Adam optimizer and trained for a total of 20 epochs. The convergence of the model is ensured by examining the loss and classification accuracies.

### 5. RESULTS

#### 5.1. Offline Model vs. Proposed Online Model

We first validate the effectiveness of the proposed online model described in section 3.4 against the baseline offline model (Section 3.1). Figure 3 plots the results, i.e., Accuracy of the baseline offline model versus the proposed online models over varying utterance lengths. Three versions of the proposed online models are trained with varying number of utterance sequences (3, 5 & 10) and evaluated for utterance sequences of lengths 1, 3, 5 and 10. The oracle end-of-utterance is assumed during the evaluation. From the plot, we observe that the accuracy of the baseline offline model is maximum for offline decoding (utterance sequence length of 1) and drops with the increase in utterance sequence lengths. Whereas, the performance of the online models is slightly lower for utterance sequence length of 1 compared to the baseline model, but exhibits less degradation with increasing length of utterance sequences. This validates the proposed online model for increment online SLU. An important observation is that the model trained on utterance sequence length of 3 performs just as well, generalizing to higher length sequences. Thus, we will only consider online models trained on utterance sequence length 3 from here onwards.
5.2. EOS Evaluations

The binary classification results of EOS detection RNN model described under Section 5.3 is presented in Table 1 row 1. The EOS system performs consistently over varying utterance sequence lengths and achieves an accuracy of approximately 92%.

Further, evaluations of the online intent module is performed by replacing the oracle sentence boundary labels by predicted outputs of the EOS module and computing the accuracy. However, there are possibilities of false positives and false negatives during EOS detection, hence, we also compute the accuracy of the intent classification when EOS predictions match with Oracle EOS. The results of the online model is presented in Figure 4 with the three accuracies over varying utterance sequence lengths. Comparing with the offline version (see Figure 3), the online version outperforms the offline SLU by a large margin evaluated on predicted EOS boundaries whenever the utterance sequence length is greater than 1. This underlines the practicality of proposed online SLU in conjunction with EOS predictor for incremental online processing. However, for an utterance sequence length of 1, the offline version is more accurate.

5.3. Multi-task Frameworks

To bridge the gap between offline SLU and the online system, we experiment with multi-task learning frameworks, described in Section 3.4 and 3.5. From the results presented in Figure 4, it is apparent that the multi-task frameworks provide better accuracies. Comparing the vanilla multi-task model (see Section 3.4) and the feed-back version (see Sections 3.5), the feed-back version achieves better accuracies for utterance sequence lengths of 1 and 3. We believe this is because the feed-back version is more sensitive to training data which was limited to utterance sequence lengths of 3. An important observation is that with the multi-task framework, especially the feedback version, the performance approaches that of the offline SLU system (96.18%), for the utterance sequence length of 3 (predicted EOS accuracy = 95.06%; Oracle&EOS accuracy = 96.1%).

5.4. Early Prediction of Intents

One of the advantages of the online SLU is the potential to predict the correct intent before the EOS is reached. To evaluate the prospects of this, we also calculated the accuracy of intent predictions during false positives of the EOS detection. Figure 5 illustrates the accuracies evaluated during (i) true negatives and true positives (i.e., matched with Oracle EOS), and (ii) false positives. We observe that the accuracy of intent detection at false positives to be high and close to the ones evaluated at Oracle EOS. This finding implies that the intent detection is accurate even before the EOS is reached, thereby hinting at possibilities of early prediction.

Additionally, we analyze the earliest time (in terms of number of words) the network starts predicting the correct intent. We obtain the early detection distribution normalizing over utterance length and then over class distribution (illustrated in Figure 6). The x-axis corresponds to normalized utterance length (0.0 being start of utterance and 1.0 being EOS) and the heat-map correspond to the number of utterances. We observe an acute peak at 1.0 for offline system, suggesting the majority of intents are detected correctly only at EOS. However, for the proposed system, the peak at 1.0 is less pronounced and the distribution is concentrated relatively more towards smaller values (left), thus suggestive of earlier detections. We find that the early detections with our proposed methods are statistically significant ($p < 0.005$) compared to offline system. Moreover, the proposed multi-task model is capable of earlier detections (statistically significant, $p < 0.05$) compared to the proposed online model.

6. CONCLUSION

In this paper, we motivated the need for incremental and online processing for SLU tasks. The low latency profile, prospects of early intent detection, independence from ASR end-pointing makes the approach attractive. We define the incremental online SLU task as real-time spoken dialog intent inference on a continuous streaming sequence of utterances with no sentence demarcation. We demonstrate that the typical offline approaches to SLU are unsuitable for incremental online processing. Multiple approaches to online SLU are proposed based on RNN intent classification. For determining the sentence boundaries, EOS detection is performed. Multi-task learning is proposed to better model the EOS and the intent detection jointly. Final results of the proposed techniques are indicative of performance approaching the accuracies of an offline SLU.

In the future, we would like to make evaluations on top of a real-time online ASR both in terms of latency profiling and accuracy profiling. The impact of ASR errors for offline versus the proposed system is worthy of investigation. Most importantly, we would like to compare the performance between the ASR end-pointing based SLU versus the proposed incremental processing. Finally, we intend to employ more complex neural network architectures and extend the framework to other SLU tasks like slot-filling, named-entity detection and assess the impact of incremental online processing. Developing heuristics for early prediction of spoken language intent is also an area of interest.

| Model          | Accuracy |
|----------------|----------|
|                | utt = 1  | utt = 3  | utt = 5  | utt = 10 |
| EOS            | 92.19    | 92.02    | 91.88    | 91.82    |
| Multi-task     | 92.39    | 92.03    | 91.86    | 91.85    |
| Multi-task FB  | 92.42    | 91.95    | 91.90    | 91.78    |

Table 1. End-Of-Sentence Detection Accuracy
