A Streaming End-to-End Framework For Spoken Language Understanding

Nihal Potdar^{*1} , **Anderson R. Avila**^{2} , **Chao Xing**^{2} , **Dong Wang**^{3} , **Yiran Cao**^{*1} , **Xiao Chen**^{2}

^{1}University of Waterloo
^{2}Huawei Noah's Ark Lab
^{3}Tsinghua University

{npotdar, y267cao}@uwaterloo.ca, {anderson.avila, xingchao.ml, chen.xiao2}@huawei.com,
wangdong99@mails.tsinghua.edu.cn

Abstract

End-to-end spoken language understanding (SLU) has recently attracted increasing interest. Compared to the conventional tandem-based approach that combines speech recognition and language understanding as separate modules, the new approach extracts users' intentions directly from the speech signals, resulting in joint optimization and low latency. Such an approach, however, is typically designed to process one intention at a time, which leads users to take multiple rounds to fulfill their requirements while interacting with a dialogue system. In this paper, we propose a streaming end-to-end framework that can process multiple intentions in an online and incremental way. The backbone of our framework is a unidirectional RNN trained with the connectionist temporal classification (CTC) criterion. By this design, an intention can be identified when sufficient evidence has been accumulated, and multiple intentions can be identified sequentially. We evaluate our solution on the Fluent Speech Commands (FSC) dataset and the intent detection accuracy is about 97 % on all multi-intent settings. This result is comparable to the performance of the state-of-the-art non-streaming models, but is achieved in an online and incremental way. We also employ our model to a keyword spotting task using the Google Speech Commands dataset and the results are also highly promising.

1 Introduction

Spoken language understanding (SLU) aims at extracting structured semantic representations, such as intents and slots, from the speech signal [Mhiri *et al.*, 2020]. For instance, a speech waveform carrying out the sentence "Lights on in the bedroom" can be represented as {intent:'activate lights' , location: 'bedroom'} (see Figure 1). Given the proliferation of voice user interfaces (VUIs) in smartphones and other gadgets, these representations are crucial to provide voice control capabilities to these devices, enabling speech as a primary mode of human-computer interaction (HCI). SLU systems are particularly useful for hands-free conversational interactions, such as while driving a car, cooking or accessing a smart home system.

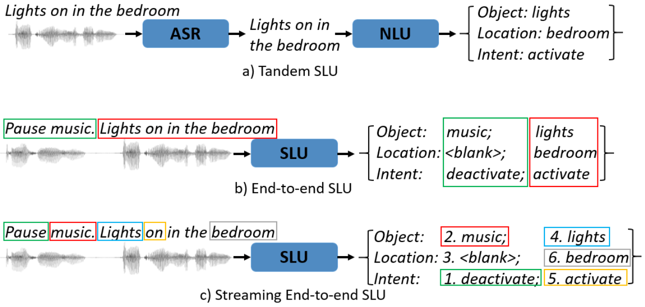


Figure 1: Intent classification based on (a) a tandem SLU system, comprised of automatic speech recognition and natural language understanding, capable of processing only a single intent at a time; (b) an end-to-end SLU that classifies multiple intents from the speech signal, enabling multi-turn dialogue; (c) a streaming end-to-end SLU which incrementally predicts intents while processing incoming chunks of the speech signal.

Tandem SLU solutions rely on the text transcription generated by an automatic speech recognition (ASR) module. The ASR output is then processed by a natural language understanding (NLU) system, responsible for interpreting the transcript content [Qian *et al.*, 2019; Shivakumar *et al.*, 2019]. This approach is represented in Figure 1 (a). A notable disadvantage of this approach relates to the fact that the ASR and the NLU systems are optimized separately, therefore not globally optimal for the target SLU task. Moreover, the NLU is often optimized on clean text while the ASR output usually contains errors. This causes severe error accumulation and amplification, aggravating the overall SLU performance [Mhiri *et al.*, 2020]. Although joint training approaches have been proposed, e.g., for the language models [Liu and Lane, 2016], the tandem nature prevents a full joint optimization.

Recently, end-to-end SLU has been proposed to tackle this problem. As shown in Figure 1 (b), the new approach extracts users' intentions directly from the speech signals, resulting in joint optimization [Haghani *et al.*, 2018; Palogiannidi *et al.*, 2020; Lugosch *et al.*, 2019; Chen *et al.*, 2018; Mhiri *et al.*, 2020]. Despite the promising progress, almost

[*] The work was done during a coop internship in Noah's Ark Lab.

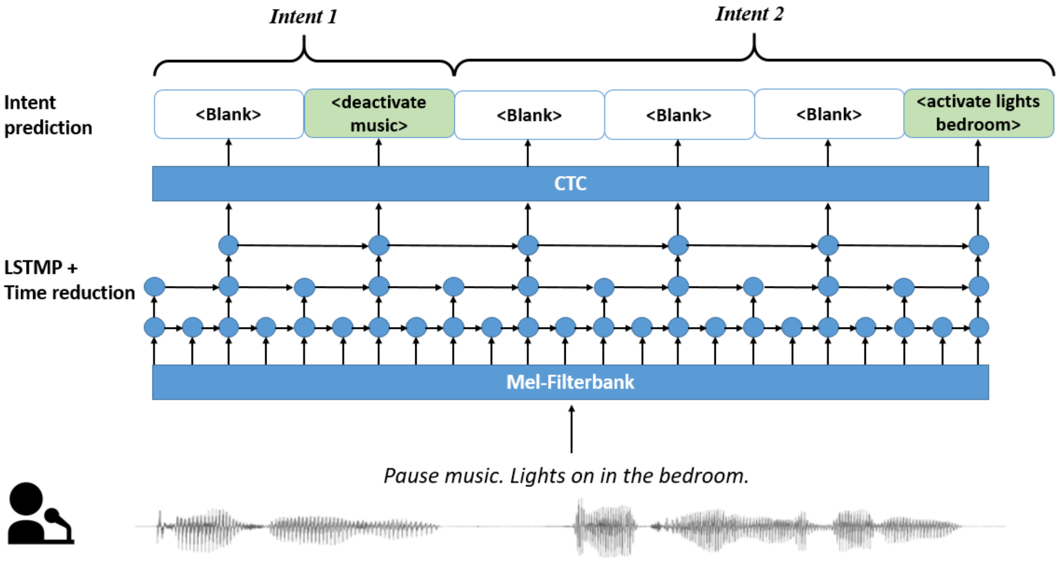


Figure 2: The proposed RNN-CTC based model enables the output to be updated in real-time as sufficient evidence is accumulated from the speech signal, hence allowing the identification of multiple intentions.

all the existing end-to-end SLU methods need to wait at least until all the speech segments are filled or the end of the speech utterance is detected by other models before starting the intent prediction, thus suffering from significant latency [Mhiri *et al.*, 2020; Haghani *et al.*, 2018]. This latency will lead to delayed feedback during human-computer interactions, which often seriously impacts users' experience [Porcheron *et al.*, 2018]. For example, in [Kohrs *et al.*, 2016], the authors conduct a systematic investigation based on functional Magnetic Resonance Imaging (fMRI) showing that delayed feedback (noticeable around 327.2-ms ±89.7) can affect the behaviour, the psychophysiology as well as the neural activity of the user. In [Schoenenberg *et al.*, 2014], it is shown that users in far-end conversation are much less attentive if the delay is more than 400 ms. To reduce the latency, some recent researches are proposed to process the speech signal segment by segment [Mhiri *et al.*, 2020], however the intent prediction is still based on the complete utterance.

Another deficiency of the existing end-to-end SLU methods is that they are short in dealing with multiple intentions. Although users can be asked to speak simply and show one intention, people tend to speak more complex utterances with multiple intents in a single round of interaction. For example, people may say 'I want to book a flight and if possible, please reserve a receiving taxi for me'. For most existing end-to-end approach, the system can only deal with one intent, perhaps the more significant one 'reserve taxi'. This will invoke additional interactions to deal with the missing intent.

In this paper, we propose a fully streaming end-to-end framework for SLU, as shown in Fig. 1 (c). The meaning of 'streaming' is two-fold: (1) the intent is identified 'online', which means one intention is identified whenever sufficient evidence has been accumulated, without the need to wait until the end of the utterance, which significantly reduce the latency; (2) the intent is identified 'incremental', which means multiple intentions can be identified sequentially. To the best of the authors' knowledge, this is the first streaming end-to-end architecture for the SLU task.

To achieve that, a unidirectional recurrent neural network (RNN), trained with the connectionist temporal classification (CTC) method, is proposed. The framework is evaluated on three underline tasks. First, our model is assessed on single-intent classification and experiments are conducted on the Fluent Speech Commands corpus (FSC) [Lugosch *et al.*, 2019]. Second, to evaluate the proposed framework on multi-intent settings, we generate versions of the FSC dataset containing speech samples with multiple intents. Details about the data generation is given in Section 4. Third, on keyword spotting, where the task refers to detecting short speech commands [Warden, 2018]. For that, we use two versions of the Google Speech Commands (GSC) dataset [Warden, 2018].

The remainder of this paper is organized as follows. In Section 2, we review the related work on end-to-end SLU and approaches to reduce latency. Section 3 presents the proposed framework. Section 4 describes our experimental setup and Section 5 discusses our results. Section 7 gives the conclusion and future works.

2 Related Work

Our work is based on the end-to-end SLU approach. Compared to the tandem ASR + SLU architecture, the end-to-end approach identifies users' intent from speech directly, hence avoiding the notorious error propagation through individual modules. A key problem of the end-to-end approach is the limited training data, due to the great variety of the speech signal when expressing an intent. Researchers have made extensive investigation and achieved promising progress. For example, in [Haghani *et al.*, 2018], several end-to-end SLU encoder-decoder solutions are investigated.

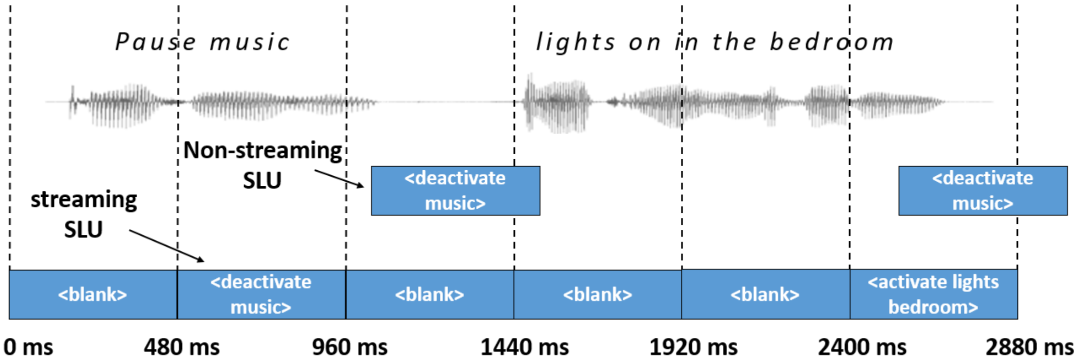


Figure 3: Illustration of non-streaming SLU and streaming SLU. Non-streaming SLU collects the speech frames of the entire utterance and predicts an intent at the end. In contrast, streaming SLU predicts the intent whenever the evidence for the intent is sufficient. Therefore, streaming SLU may predict an intent before the utterance is end, and can predict multiple intentions in a sequential way.

In [Palogiannidi *et al.*, 2020], an end-to-end RNN-based SLU solution is proposed. Several architectures for predicting intents conditioned to the slot filling task are introduced. The authors demonstrate that data augmentation is required in order to attain the state-of-the-art performance while training deep architectures for end-to-end SLU. Another recent study introduces the Fluent Speech Command dataset and also presents a pre-training methodology for end-to-end SLU models. Their approach is based on adopting ASR targets, such as words and phonemes, that are used to pre-train initial layers of their final model. These classifiers, once trained, are discarded and the embeddings from the pre-trained layers are used as features for the SLU task. The authors show improved performance on large and small SLU training sets with the proposed pre-training strategy [Lugosch *et al.*, 2019]. A similar approach is found in [Chen *et al.*, 2018]. The authors also propose to fine-tune the front layers of an end-to-end CNN-RNN model that learns to predict graphemes. This pre-trained acoustic model is optimized with the CTC loss, which allows the model to learn the alignments between input and output sequences. This acoustic model is then combined with a semantic model to predict intents. Note that the CTC loss is calculated for the ASR module, not for the SLU model as in our work.

Treatments of latency have been investigated in the tandem ASR+NLU architecture, in principle based on segment-level ASR output. For example, [Shivakumar *et al.*, 2019] propose a recurrent neural network (RNN) that processes ASR output of arbitrary length and provides intent predictions in real-time. Similarly, [Liu and Lane, 2016] uses an RNN to perform online intent detection and slot filling jointly as ASR output arrives. For the end-to-end system, [Mhiri *et al.*, 2020] propose a speech-to-intent model based on a global max-pooling layer that allows for processing speech sequentially while receiving upcoming segments. Although the authors are able to achieve low latency in their results, the solution is not online, as the intent prediction still requires the entire sentence to be processed completely.

Finally, incremental multiple intention detection has been studied with the ASR+NLU architecture [Shivakumar *et al.*, 2019], based on segmental ASR output and online and incremental NLU. For the end-to-end architecture, no extensive research has been found.

Compared to the aforementioned research, our proposal is a complete streaming end-to-end SLU framework, capable of incrementally predicting multiple intents from incoming chunks of the speech signal. Next, we shall give the reader details on our proposed solution.

3 Proposed Framework

We start this section by presenting the main ideas and challenges behind modeling a streaming end-to-end SLU solution. Next, we introduce the proposed framework. We show how CTC optimization can help our framework to perform multi-intent classification without requiring prior alignment knowledge between the input and output sequences. We finish this section by giving details on how to train our proposed framework with a two-stage pre-training approach.

3.1 Principle

End-to-end SLU. An ASR-free end-to-end SLU model aims at classifying a speech utterance, represented by a sequence of acoustic features X, as a single semantic label L (or intent class), hence bypassing the need of an explicit combination of an ASR and an NLU modules [Qian *et al.*, 2019]. This is achieved by maximizing the following posterior probability,

$$L^* = \arg\max_L P(L|X, \theta) \tag{1}$$

where θ is the parameters of the end-to-end neural network model. Compared to the tandem ASR + NLU architecture, the end-to-end approach solves the error propagation problem, but usually requires large amounts of data to deal with the high variation in speech signal when expressing the same intent. Methods including data augmentation, pre-training and multi-task learning have been widely used [Palogiannidi *et al.*, 2020; Lugosch *et al.*, 2019; Chen *et al.*, 2018].

End-to-end streaming SLU. The above end-to-end approach predicts the intent with the entire utterance, and can only predict one intent for a single utterance. This approach is non-streaming. For a streaming approach, however, the model is expected to perform a sequence-to-sequence prediction, which predicts an intent when the evidence is sufficiently accumulated, and multiple intents can be predicted incrementally. The reader can refer to Figure 3 as illustration of the difference between the non-streaming and streaming approaches. Compared to the non-streaming case, the primary difficulty for the streaming model is how to accumulate evidence for each intent so that appropriate intents can be 'fired' when the evidence is sufficient. In the model that will be presented shortly, we employ CTC training as the major tool to solve this problem.

3.2 Proposed Architecture

We formulate the streaming end-to-end SLU as a sequence-to-sequence task, where our goal is to predict the sequence of intents L by given an input sequence of acoustic features X. The proposed architecture is shown in Fig. 2. The main structure of the model is a unidirectional RNN network, where the units are long short-term memory with projection (LSTMP). A potential problem for SLU is that the lengths of the speech input and the intent labels are significantly different. The same problem arises in ASR, but for SLU it is more evident as an intention often corresponds to a much longer speech segment than a word. In order to condense the semantic information and reduce the length discrepancy, time reduction [Chan *et al.*, 2015] was employed to reduce the frame rate for each layer.

3.3 CTC Criterion

The key problem of the proposed streaming end-to-end architecture is how to fire intentions *automatically* when evidence is sufficient. We resort to the Connectionist Temporal Classification (CTC) training. The CTC training was first motivated to train RNNs of unsegmented data [Graves *et al.*, 2006]. Previous to CTC, training RNNs required prior segmentation of the input sequence. With such segmentation, each feature was labelled with an appropriate target and the RNN was trained to predict the target step by step, via the back-propagation through time (BPTT). With CTC, all the potential alignments between the input and output sequences are considered during the training, by inserting blank tokens (<blank>) between any output labels (including the beginning and end of the sequence) to ensure the input and output sequences have the same length. This approach has been successfully employed in various tasks including speech recognition [Miao *et al.*, 2016] and machine translation [Libovický and Helcl, 2018].

Formally, the conditional probability of a single alignment, α, is the product of the probabilities of observing α_t at time t and can be represented as

$$P(\alpha|X,\theta)=\prod_{t=1}^{T}P(\alpha_t|X,\theta) \tag{2}$$

where α_t represents the given label (including the blank token) at time t, according to the alignment α. Because $P(\alpha|X)$ defines mutually exclusive alignment, the conditional probability for the target sequence Y given the input sequence X amounts to the sum of the probabilities associated with all the alignments:

$$P(L|X,\theta)=\sum_{\alpha\in\mathcal{B}_{X,Y}}\prod_{t=1}^{T}P(\alpha_t|X,\theta) \tag{3}$$

where $\mathcal{B}_{X,L}$ is the set of all valid alignments. The CTC loss is then defined as

$$L_{CTC}(X,L)=-\log\sum_{\alpha\in\mathcal{B}_{X,L}}\prod_{t=1}^{T}P(\alpha_t|X,\theta) \tag{4}$$

This model can be efficiently trained following a forward-back algorithm [Graves *et al.*, 2006]. A favorable feature of a CTC model is that during decoding, true target labels are output only the evidence for them are sufficiently accumulated, otherwise the model will output the blank token [Sak *et al.*, 2015]. Once a label has been output, its evidence is reset and a new round of evidence accumulation is started. This feature perfectly fits the request of the streaming SLU, where we expect the model to fire an intent if sufficient evidence from the speech signal has been accumulated, and once it is fired, the evidence should be accumulated again. By this setting, multiple intents can be sequentially identified, no matter how long the speech signal is.

3.4 Two-stage Pre-training

Although CTC can potentially solve the streaming SLU task, some training skills are still required in order to tackle the intrinsic data sparsity problem (as in non-streaming models). For that, we designed a two-stage pre-training scheme as described below. Our experiments show that the pre-training methods are crucial for training a high-quality streaming end-to-end model.

ASR pre-training. In the first stage, we initialize the feature extraction layers with a pre-trained ASR model. Only the first layer was used for this purpose. After training, the weights were kept frozen and used to provide embeddings for the next LSTM layers. The upper LSTM layers use these embeddings (or features) to make predictions. This procedure aims at leveraging the large speech corpora and the fine-grained word labels in ASR. For that, we utilize the 100 hours of speech signals in the Librispeech dataset [Panayotov *et al.*, 2015]. Our pre-trained model is based on classifying character-level targets, employing the CTC loss. Using characters instead of words as the training target simplifies the training task [Lugosch *et al.*, 2019]. In this study, we only reuse the first layer of the ASR model for the SLU model, which ideally plays the role of robust feature extraction. Different from the approach investigated in [Kuo *et al.*, 2020] and [Lugosch *et al.*, 2019], where the weights of the pre-trained ASR model were fine-tuned while training the SLU model, the layer borrowed from ASR model in our study remains frozen.

Cross-entropy pre-training. In the second stage, we pre-train the SLU model (the first layer fixed) with a cross-entropy (CE) loss, before taking the CTC training. For simplicity, we compute the CE loss at the last time step only and back-propagate to the beginning of the sequence. After this pre-training, the model is fine-tuned by using the CTC loss.

4 Experimental Setup

We evaluate the proposed model in three tasks: single-intent classification, multi-intent prediction and spoken keyword spotting. In this section, we firstly present the datasets used and then describe the experimental settings.

4.1 Datasets

Four datasets were involved in our experiments, all audio clips were (re)sampled at 16 kHz.

- All of our pretrained ASR models were trained on the LibriSpeech corpus, which was introduced in [Panayotov *et al.*, 2015]. It was derived from audiobooks and contains roughly 1000 hours of speech. We selected 100 hours of clean speech to train the ASR model. There are couple of reasons for that. First, we are proposing a compact model with relatively reduced number of parameters. Second, the pre-trained ASR is attained by optimizing the weights in the first layer, which means even fewer parameters to be optimized. Therefore, we considered 100 hours (3x more than the in-domain FSC data) enough to train the first layer. Note that this layer is kept frozen while training the rest of the network. Empirically, we found no benefit in unfreezing the first layer while training our model with in-domain data. Similar findings have been found for the same dataset in previous work [Lugosch *et al.*, 2019].
- The Fluent Speech Commands (FSC) dataset [Lugosch *et al.*, 2019] was used to train and evaluate our SLU model for intent classification. It comprised single-channel audio clips collected using crowd sourcing. Participants were requested to speak random phrases for each intent twice. The dataset contained about 19 hours of speech, providing a total of 30,043 utterances spoken by 97 different speakers. Each utterance contains three slots: action, object, and location. We considered a single intent as the combination of all the slots (action, object and location), resulting 31 intents in total. The whole dataset was split to three subsets: the training set (FSC-Tr) contained 14.7 hours of data, totalling 23,132 utterances from 77 speakers; the validation set (FSC-Val) and test set (FSC-Tst) comprised 1.9 and 2.4 hours of speech, leading to 3,118 utterances from 10 speakers and 3,793 utterances from other 10 speakers, respectively. The validation set was used to select the model, while the test set was used to evaluate the SLU performance.
- To train and test the multi-intent prediction model, we generated three additional multi-intent datasets using the utterances from the FSC dataset. The first dataset, namely FSC-M2, was attained by concatenating two utterances from the same speaker into a single sentence. We have made sure that all the possible intent combinations were evenly distributed in the dataset. The training part of FSC-M2 (FSC-M2-Tr) contained 57,923 utterances (74.27 hours) selected from the FSC training data, and the test part (FSC-M2-Tst) contained 8,538 utterances (11.80 hours) from the FSC test data. Following a similar procedure, we constructed a three-intent dataset (FSC-M3) by augmenting the utterances in FSC-M2 with a new utterance from FSC. Finally, we combined roughly 40 % of random samples from FSC, FSC-M2, and FSC-M3 to build a large multi-intent set (FSC-MM), where the data splitting for training and test was the same as the original datasets. For model selection, we also constructed validation sets for each multi-intent condition.
- We also used the Google Speech Commands (GSC) dataset [Warden, 2018] to verify the capability of our model in spotting short key words. The provider of the dataset published two versions. Version 1 comprised 64,727 utterances from 1,881 speakers, a total of 30 words. We split the dataset into training, validation and test sets, containing 14.08, 1.88, 1.89 hours of speech signals respectively. Version 2 was larger, consisting of 105,829 utterances of 35 words, recorded by 2,618 speakers. The training, validation and test sets contain 23.30, 3.02, 2.74 hours of speech, respectively.

4.2 Experimental Settings

In this work, audio signals are sampled at 16 kHz. The Kaldi toolkit is used to extract 80-dimensional log Mel-Filterbank features with 3-dimensional pitch, totalling 83 dimensions. To extract the Mel features, the audio signal is processed in frames of 320 samples (i.e., 20-ms window length), with a step size of 160 samples (that is, 10-ms hop-size). Global Cepstral Mean and Variance Normalization (CMVN) are applied in order to mitigate the mismatch between training and testing data.

The model was trained using the ADAM optimizer [Loshchilov and Hutter, 2017], with the initial learning rate set to 0.0001. Dropout probability was set to 0.1 and the parameter for weight decay was set to 0.2. For the ASR pre-training, the ASR model was trained 100 epochs; for the CE pre-training, the model was trained for 10 epochs with the CE criterion.

5 Experimental Results

We present the results on three tasks: single-intent classification, multi-intent prediction and spoken keyword spotting. For the ablation study, note that we present the results with and without the two pre-training techniques. Results are all reported on test sets.

Model	Streaming	Accuracy
RNN+Pre-training [Lugosch *et al.*, 2019]	no	98.8
CNN+Segment pooling [Mhiri *et al.*, 2020]	no	97.8
CNN+GRU(SotA) [Tian and Gorinski, 2020]	no	99.1
RNN+CTC	yes	12.66
+ ASR Pre-training	yes	15.83
+ CE Pre-training	yes	98.90

Table 1: Experimental results on FSC for single-intent classification. Performance is reported in terms of accuracy (%).

5.1 Experiment I: Single-intent Classification

The first experiment is single-intent classification, conducted with the FSC dataset. The model has three LSTMP layers, each with 1024 hidden states followed by a projection layer with 320 dimension. As described in Section 3.4, the first layer is based on a pre-trained ASR model where the current frame is stacked with 7 frames to the left. Frame-skipping is employed, with only 1/3 frames need to be processed, leading to faster computation. The next two LSTMP layers involve a time reduction operation with the reduction factor 4. This results in a 480-ms input context for each output time step. The output of the third LSTMP layer is fed to two fully connected layers and the final output corresponds to the 31 intents.

Table 1 presents the results for this task. For comparison, three recent end-to-end SLU solutions are adopted as benchmarks: the RNN model with pre-training presented by [Lugosch *et al.*, 2019], the CNN model with segment pooling presented by [Mhiri *et al.*, 2020], and the CNN trained with Reptile presented by [Tian and Gorinski, 2020]. The CNN+Reptile approach represents the state-of-the-art (SotA) performance on this task. We emphasize that all of the three models are non-streaming and so cannot perform online and incremental predictions. Nevertheless, their results set good reference for our streaming model.

The results in Table 1 demonstrated that our proposed streaming model is highly effective: it achieves performance as high as 98.90 % with the two-stage pre-training. This result outperforms the first two benchmarks, and is very close to the SotA model [Tian and Gorinski, 2020], while our approach is streaming.

Focusing on the pre-training methods, it can be observed that both the ASR pre-training and the CE pre-training improve the performance, though the contribution of the CE pre-training is much more significant. This is not surprising as the supervision from the target (intent) is very sparse and usually appears at the end of an utterance. The CE pre-training offers a strong bias on the more reasonable alignments for the CTC training.

An interesting observation is that our streaming model usually predicts the intent before the end of the utterance. In most cases, the spotting position is at the 2/3 length of the utterance. This implies that the model has learned to accumulate the evidence and can make the prediction decision if the evidence has been strong enough, rather than waiting for the end of the utterance. This feature is highly desirable. It enables a truly online SLU, and it is a major advantage that the CTC model possesses.

		Trained			
		FSC-Tr	FSC-M2-Tr	FSC-M3-Tr	FSC-MM-Tr
Tested	FSC-Tst	**98.90**	98.77	97.96	**98.90**
	FSC-M2-Tst	56.90	**97.93**	96.90	97.89
	FSC-M3-Tst	26.73	97.15	97.01	**97.28**

Table 2: Experimental results for the multi-intent prediction. Performance is reported in terms of accuracy (%).

Model	V1	V2
Att-RNN [Andrade *et al.*, 2018]	95.6	96.9
TC-ResNet [Choi *et al.*, 2019]	-	96.6
SincConv+DSConv [Mittermaier *et al.*, 2020]	96.6	97.4
TENET12+MTConv [Li *et al.*, 2020]	96.84	-
CTC	62.64	62.98
+ Pre-trained ASR	64.16	63.59
+ Pre-trained CE	96.35	97.10

Table 3: Experimental results on the keyword spotting task. Performance is reported in terms of accuracy (%) on the GSC dataset V1 and V2.

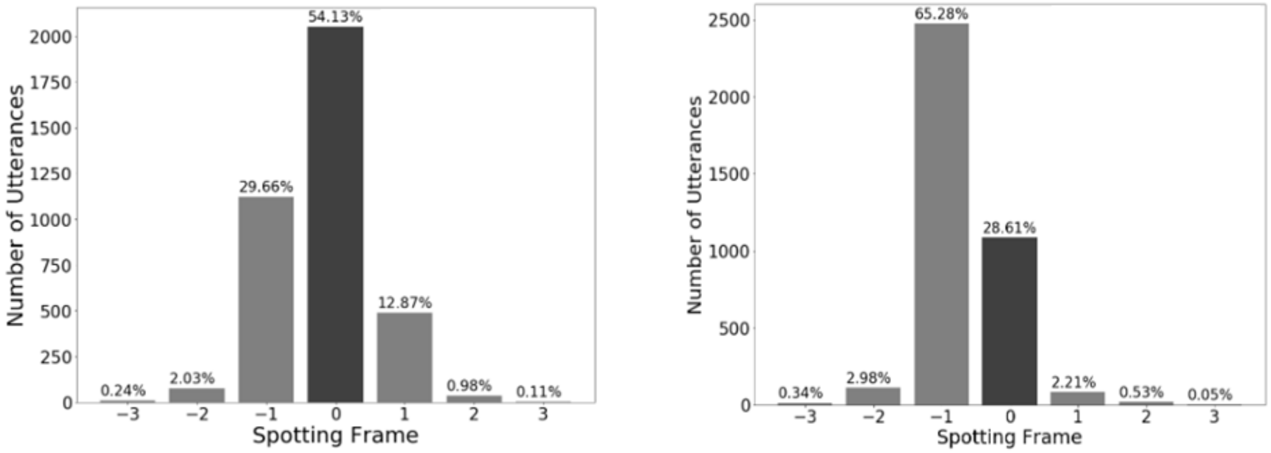
5.2 Experiment II: Multi-intent Prediction

In the second experiment, we evaluate our model's ability to predict multiple intents. For that, we adopt our architecture based on the two-stage pre-training approach, which provides the best results in the first experiment. We first use the model trained on the single-intent corpus (FSC-Tr) to predict multiple intents, and then retrain multi-intent models using multi-intent corpora. Note that the benchmark models in the previous experiment cannot predict multiple intents, so we did not report them here.

The performance is presented in Table 2, where each row represents a test condition and each column represents a training condition. Note that for the multi-intent test, only if all the intents involved in the utterance are identified, the result is recognized as correct. Therefore, the multi-intent prediction is much more challenging than the single-intent classification.

Table 2 shows that, when trained with single-intent data and tested on two-intent and three-intent utterances, the performance of our streaming model is significantly degraded, specially in the three-intent case. This is mainly because there is no prior knowledge about the intent boundaries involved in the training data. Nonetheless, the single-intent model can still predict multiple intents in some cases, especially in the two-intent setting. This is surprising and demonstrated that the CTC streaming model is rather powerful and has learned how to accumulate evidence for intents and reset it when an intent has been detected.

When multi-intent data are employed to train the model, the performance on the two-intent and three-intent conditions is significantly improved. Surprisingly, the accuracy of the multi-intent prediction is almost the same as the single-intent prediction, although the former is much harder. This implies that by learning the multi-intent samples, the model has well learned how to reset the evidence quickly, so that it can prepare for the next intent prediction in time. Remarkably, the model seems to learn this knowledge with two-intent data only and this knowledge can be generalized to condi-



(a)

(b)

Figure 4: Relative spotting positions compared to ground truth, with (a) model trained and evaluated on single intent experiment and (b) model trained on multi-intents setting and evaluated first intent. The x-axis represents the relative distance between the spotting position and the ground truth position. Negative values refer to early spotting and positive values refer to late spotting.

tions with more intentions. For example, on the three-intent test, the accuracy was increased from 26% to 97% when we trained the model with two-intent data.

5.3 Experiment III: Keyword Spotting

In this last experiment, we customize the streaming end-to-end architecture and apply it to a keyword spotting task. This model consists of two LSTMP layers. The first LSTMP layer involves 1024 hidden states, followed by a projection layer with 320 units. This layer is initialized by ASR pre-training as in the intention prediction experiment. The second LSTMP layer is similar, but time reduction is applied, so that the time scope is enlarged by 4 times. The output from the second LSTMP layer is fed into two fully connected layers, and the output corresponds to the keywords for detecting.

Two models were trained using the training set of GSC V1 and V2 respectively, and were tested on the test set of GSC V1 and V2 individually. For the test on each dataset, 12 keywords were selected. The performance in terms of detection accuracy is presented in Table 3. For comparison, we reported the results of four benchmarks that represent the state-of-the-art research on this dataset. Results show that on both the two test sets, our model based on two-stage pre-training offers competitive performance when compared to the benchmark results. Meanwhile, our model is streaming and holds the great advantage on spotting multiple keywords online and incrementally.

6 Analysis for Early Spotting

A key advantage of our streaming SLU system is that intents or keywords can be determined whenever evidence is sufficient. In this section, we present some statistical analysis of such early-spotting behavior based on the intent detection. The first analysis investigates the statistics of relative position of the spottings, i.e., the distance between the spotting positions and ground truth positions. Note that we treat the last non-silence frame corresponding to an intention as its ground truth position, which can be determined by Voice Activity Detection (VAD). Figure 4a shows the relative positions of the spottings in the intent detection experiments, where a negative distance denotes an early spotting, and a positive distance denotes a late spotting. It can be found that in the single-intent scenario, more than 30% of the spottings occur earlier than the ground truth positions, and this proportion is 68% in the multi-intent scenario. The second analysis investigates the relation between the early-spotting behavior and the length of the sentence. The results are shown in Figure 5, where the x-axis represents the sentence length in characteres, and the y-axis represents the relative spotting position. It can be found that long sentences are more likely to be early spotted than short sentences.

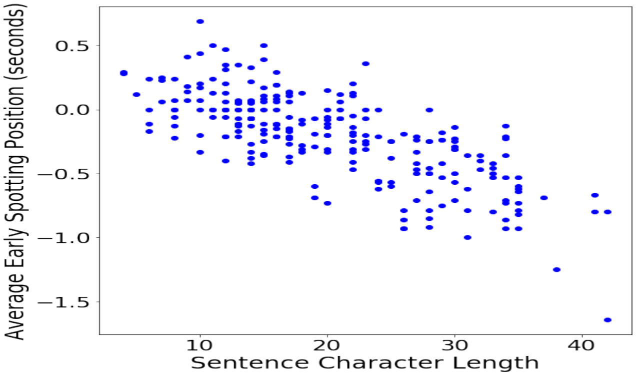


Figure 5: The relation between the early-spotting behavior and the sentence length in characters. The x-axis represents the sentence length, and the y-axis represents the relative spotting position.

7 Conclusion

In this paper, we propose an end-to-end streaming SLU framework, which can predict multiple intentions in an online and incremental way. Our model is based on a unidirectional LSTM architecture, optimized with the alignment-free CTC loss and pre-trained with the cross-entropy criterion. We test the model on the intent prediction task. Results show that our model achieves comparable performance to start-of-the-art non-streaming models for single-intent classification. For multi-intent prediction, which is more challenging and impossible for the existing non-streaming models, our approach almost maintains the accuracy of single-intent classification. The model was also tested on a keyword spotting task, and the performance is competitive compared to four benchmarks. As future work, we aim to explore streaming intent + slot classification, and integrate local attention mechanism into our streaming solution.

References

[Andrade *et al.*, 2018] Douglas Coimbra Andrade, Sabato Leo, Martin Loesener Da Silva Viana, and Christoph Bernkopf. A neural attention model for speech command recognition. *arXiv preprint arXiv:1808.08929*, 2018.

[Chan *et al.*, 2015] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

[Chen *et al.*, 2018] Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore. Spoken language understanding without speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE, 2018.

[Choi *et al.*, 2019] Seungwoo Choi, Seokjun Seo, Beomjun Shin, Hyeongmin Byun, Martin Kersner, and Beomsu Kim. Temporal convolution for real-time keyword spotting on mobile devices. *arXiv preprint:1904.03814*, 2019.

[Graves *et al.*, 2006] Alex Graves, Santiago Fernández, and Faustino Gomez. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[Haghani *et al.*, 2018] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, and Neeraj Gaur. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726, 2018.

[Kohrs *et al.*, 2016] Christin Kohrs, Nicole Angenstein, and André Brechmann. Delays in human-computer interaction and their effects on brain activity. *PloS one*, 11(1):e0146250, 2016.

[Kuo *et al.*, 2020] Hong-Kwang J Kuo, Zoltán Tüske, Samuel Thomas, and Yinghui Huang. End-to-end spoken language understanding without full transcripts. *arXiv preprint:2009.14386*, 2020.

[Li *et al.*, 2020] Ximin Li, Xiaodong Wei, and Xiaowei Qin. Small-footprint keyword spotting with multi-scale temporal convolution, 2020.

[Libovickỳ and Helcl, 2018] Jindřich Libovickỳ and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv preprint arXiv:1811.04719*, 2018.

[Liu and Lane, 2016] Bing Liu and Ian Lane. Joint online spoken language understanding and language modeling with recurrent neural networks. *arXiv preprint arXiv:1609.01462*, 2016.

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

[Lugosch *et al.*, 2019] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*, 2019.

[Mhiri *et al.*, 2020] Mohamed Mhiri, Samuel Myer, and Vikrant Singh Tomar. A low latency asr-free end to end spoken language understanding system. *arXiv preprint arXiv:2011.04884*, 2020.

[Miao *et al.*, 2016] Yajie Miao, Mohammad Gowayyed, Xingyu Na, Tom Ko, and Florian Metze. An empirical exploration of ctc acoustic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2623–2627. IEEE, 2016.

[Mittermaier *et al.*, 2020] S. Mittermaier, L. Kürzinger, B. Waschneck, and G. Rigoll. Small-footprint keyword spotting on raw audio data with sinc-convolutions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7454–7458, 2020.

[Palogiannidi *et al.*, 2020] Elisavet Palogiannidi, Ioannis Gkinis, George Mastrapas, Petr Mizera, and Themos Stafylakis. End-to-end architectures for asr-free spoken language understanding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7974–7978. IEEE, 2020.

[Panayotov *et al.*, 2015] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[Porcheron *et al.*, 2018] Martin Porcheron, Joel E Fischer, Stuart Reeves, and Sarah Sharples. Voice interfaces in everyday life. In *proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[Qian *et al.*, 2019] Yao Qian, Rutuja Ubale, Patrick Lange, Keelan Evanini, Vikram Ramanarayanan, and Frank K Soong. Spoken language understanding of human-machine conversations for language learning applications. *Journal of Signal Processing Systems*, pages 1–13, 2019.

[Sak *et al.*, 2015] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

[Schoenenberg *et al.*, 2014] Katrin Schoenenberg, Alexander Raake, and Judith Koeppe. Why are you so slow?–misattribution of transmission delay to attributes of the conversation partner at the far-end. *International journal of human-computer studies*, 72(5):477–487, 2014.

[Shivakumar *et al.*, 2019] Prashanth Gurunath Shivakumar, Naveen Kumar, Panayiotis Georgiou, and Shrikanth Narayanan. Incremental online spoken language understanding. *arXiv preprint arXiv:1910.10287*, 2019.

[Tian and Gorinski, 2020] Yusheng Tian and Philip John Gorinski. Improving end-to-end speech-to-intent classification with reptile. *arXiv preprint:2008.01994*, 2020.

[Warden, 2018] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.