Journal of Signal Processing Systems (2020) 92:805–817
https://doi.org/10.1007/s11265-019-01484-3

Spoken Language Understanding of Human-Machine Conversations for Language Learning Applications

Yao Qian[1] · Rutuja Ubale[1] · Patrick Lange[1] · Keelan Evanini[2] · Vikram Ramanarayanan[1,3] · Frank K. Soong[4]

Received: 15 February 2019 / Revised: 2 August 2019 / Accepted: 9 September 2019 / Published online: 11 November 2019

Abstract
Spoken language understanding (SLU) in human machine conversational systems is the process of interpreting the semantic meaning conveyed by a user's spoken utterance. Traditional SLU approaches transform the word string transcribed by an automatic speech recognition (ASR) system into a semantic label that determines the machine's subsequent response. However, the robustness of SLU results can suffer in the context of a human-machine conversation-based language learning system due to the presence of ambient noise, heavily accented pronunciation, ungrammatical utterances, etc. To address these issues, this paper proposes an end-to-end (E2E) modeling approach for SLU and evaluates the semantic labeling performance of a bidirectional LSTM-RNN with input at three different levels: acoustic (filterbank features), phonetic (subphone posteriorgrams), and lexical (ASR hypotheses). Experimental results for spoken responses collected in a dialog application designed for English learners to practice job interviewing skills show that multi-level BLSTM-RNNs can utilize complementary information from the three different levels to improve the semantic labeling performance. An analysis of results on OOV utterances, which can be common in a conversation-based dialog system, also indicates that using subphone posteriorgrams outperforms ASR hypotheses and incorporating the lower-level features for semantic labeling can be advantageous to improving the final SLU performance.

Keywords Spoken language understanding · Human-machine conversational systems · Computer assisted language learning · End-to-end modeling · Education

1 Introduction

The popularity of intelligent virtual assistants, such as Alexa (Amazon), Siri (Apple), Google Home, and Cortana (Microsoft), has accelerated the progress of human-machine conversational systems towards offering more natural, intuitive, robust, and effective interactions. The widespread use of such virtual assistants has also spurred on the development of many new and innovative applications, such as human-computer conversation-based language learning, which will be the focus of this paper. Human-machine conversation, empowered by artificial intelligence, can facilitate natural and effective learning by providing timely assessments, interactive feedback, and personalized learning materials to a student when a human teacher is not available. Spoken language understanding (SLU), the process of interpreting the semantic meaning conveyed in a spoken utterance, is a key component in enabling an interactive system to take appropriate actions in a human-computer conversation. Currently, achieving high levels of SLU performance is still a challenge in many domains, especially in the case of realistic, interactive spoken language learning applications. This is because it is difficult to obtain sufficiently large amounts of labeled data that is matched with the real user scenarios when new applications are developed from scratch, thus requiring them to be bootstrapped from unlabeled and mismatched data.

State-of-the-art SLU systems generally contain two components: the automatic speech recognizer (ASR), which decodes the input speech into text, and the natural language understanding (NLU) module, which transforms the ASR hypothesis into a concept or semantic label that can drive subsequent computer behavior. These two components are typically based on statistical models trained on a large amount of data using a variety of

✉ Yao Qian
yqian@ets.org

Extended author information available on the last page of the article.

machine learning methods. In contrast to NLU on written text, the effectiveness of the SLU system also largely depends on the performance of the ASR system, its robustness to ASR errors, and its ability to appropriately process spontaneous speech, which can contain hesitations, corrections, repetitions, etc. Exacerbating this, non-native speech collected by conversation-based language learning applications may be characterized by additional difficult traits such as pronunciation errors, large numbers of disfluencies, ungrammatical phrases, loan words, etc., which can make the ASR output even less accurate. It should be noted that even human experts often find it difficult to transcribe spoken responses with these characteristics produced by non-native speakers.

SLU can be regarded as a cascaded conversion from the speech signal input to a semantic label output in acoustic, phonetic, lexical and semantic spaces. The following design considerations can help make the SLU system robust through the successive steps in the conversion process: the acoustic model (AM) should be resistant to different acoustic channel conditions and ambient noises; the lexical model should be adaptable to speaker variability exhibited in accented pronunciations and out-of-vocabulary words (OOV); the language model (LM) should be flexible enough to handle syntactic and grammatical variation; and the NLU model should be insensitive to ASR errors and variations in pragmatics. To address these SLU robustness issues in the context of language learning applications, we investigate using an end-to-end modeling approach, which utilizes as little prior knowledge as possible by skipping one, two or all of the stages (along with the corresponding required human labels) in the cascaded conversion process, to predict semantic labels from the speech signal directly using ASR-free modeling and from the sub-phone search space by skipping the language model; furthermore, we explore how the performance can be improved by fusing the results from the different levels.

2 Related Work

Most state-of-the-art SLU systems utilize deep learning technologies to perform semantic tagging with transcriptions or ASR hypotheses [1–3]. Recurrent Neural Networks (RNNs) with a variety of different architectures have been proposed for semantic slot filling and they have typically been evaluated on the well-known Airline Travel Information System (ATIS) benchmark task. The experimental results show that the RNN-based models outperform a conditional random field (CRF) baseline [1]. Joint slot filling and intent detection based on Convolutional Neural Networks (CNNs), in which the features are extracted through CNN layers and shared by these two tasks, also leads to improved performance over the CRF baseline [2].

Many researchers have tried to skip the ASR step entirely or to only use partial information extracted from its modules for semantic classification [4–7]. Utterance classification can be performed by unsupervised phonotactic models together with token sequence classifiers [6], which can avoid manual word-level transcription of the utterances and achieve a performance close to those of conventional methods involving word-level language models. Techniques for building call routers from scratch without any knowledge of the application vocabulary or grammar have also been explored [4].

Recently, end-to-end learning directly from speech input has become popular for various spoken language processing tasks. These architectures can model the output directly from the speech signal, i.e., using spectral features such as MFCCs, filter bank features, or directly from the raw waveform. End-to-end sequence-to-sequence learning has been used in text-to-speech synthesis systems [8–10] replacing the front-end and back-end modules with a single system that converts character sequences to a Mel spectrogram representation which is then converted to speech. End-to-end models built for speaker recognition, in which feature engineering and scoring is performed in a single model, can outperform state-of-the art methods [11, 12]. End-to-end networks have also been investigated for a range of other speech processing tasks, including native language identification [13], language recognition [14], emotion recognition [15], keyword spotting [16, 17] and voice activity detection [17].

Traditional ASR systems consist of three modules that are compiled independently of each other: an acoustic model, a pronunciation model and a language model. Furthermore, a typical spoken language understanding system consists of a pipeline of components, and the ASR and NLU components require independent training. The ASR model is optimized using the word error rate (WER) criterion while the NLU model is trained to maximize classification accuracy. End-to-end learning for speech recognition has become very popular in recent years[18–24]. In an end-to-end ASR system, the individual components can be learned jointly in a single module using deep learning.

Following the success of end-to-end ASR systems, we made the first ever attempt at designing an end-to-end spoken language understanding system [25] thereby replacing the ASR and NLU components with a single component that can extract intents/semantic labels directly from the user's spoken response. While in an end-to-end ASR system the task is to learn a sequence-to-sequence representation, in an end-to-end SLU system the task is to learn

a sequence-to-class (intent/domain/semantics) representation which requires a more complex transformation of the input. We employed MFCC features as the input to a long short-term memory recurrent neural network-based (LSTM-RNN) encoder-decoder network in [25, 26]. In addition, there have been some recent attempts at end-to-end SLU using Mel filterbank features as input to different encoder-decoder architectures [27, 28]. While research efforts in [27, 28] for designing end-to-end spoken language understanding leveraged massive amounts of data to train models, our work [25, 26] specifically looks at building models in low-resource settings.

3 Human-machine Conversation-based Language Learning

This research is being conducted in the domain of human-machine conversation-based language learning, a particularly promising application area for spoken dialog system (SDS) technology. Due to the increasing demand to learn English for success in the global economy, it is difficult for English learners around the world to have sufficient opportunities to practice speaking and receive feedback about their speaking proficiency; SDS-based English learning applications can provide a means to achieve these goals when there is limited access to human instructors. The main criterion in designing the SDS-based speaking tasks is to make them as authentic as possible in order to provide learners with valid opportunities for practicing the English skills that they need in order to improve their communicative competence; therefore, the tasks considered for this effort are situation-based, goal-oriented tasks including conversational functions that are important for language learners, such as ordering food in a restaurant, interviewing for a job, making requests in a workplace environment, asking for information, etc. (an online sample of some of the conversation-based tasks is available at http://englishtasks.org).

Since the main goal of conversation-based tasks for the purpose of language learning is to provide the learners with opportunities to practice speaking, designers typically aim to maximize the user's speaking time in order to provide more practice opportunity and elicit a more valid response for scoring and feedback. This is in contrast to standard commercial SDS applications, such as automated customer support systems and digital personal assistants which typically aim to minimize user speaking time in order to complete the conversation as quickly as possible. This goal of maximizing user speaking time means that the conversations typically encourage the learners to use a broad range of vocabulary and sentence structures; this can make the task of SLU more challenging than it would be in a standard SDS application in which the user's utterances should be as constrained as possible to facilitate successful task completion.

We used the multimodal dialog system HALEF[1] to collect the data used in this study. HALEF is a modular system based on open-source components and leverages W3C recommendations and open industry standards. Further details about the architectural components are provided in [30]. The system is hosted in the cloud and users can connect to it using web browsers supporting WebRTC. This enables us to leverage crowdsourcing to collect large amounts of data to develop applications and for usage in research studies. Media and metadata from the conversations are stored on the backend and are then used for iterative improvements. The improvement process has several steps: the data is first transcribed, then annotated with semantic labels, and finally used to update and refine the conversational task design and models for speech recognition and spoken language understanding [29].

This study examines an interactive speaking task that simulates a job interview scenario. The conversation is set up as a system-initiated dialog in which a representative at a job placement agency interviews the language learner about the type of job they are looking for and their qualifications. The crowdsourcing user pool was restricted to non-native speakers of English. An example of one dialog state in the job interview task, including the question posed by the system, human transcriptions of sample user responses, and the corresponding gold-standard semantic labels for each utterance, is shown in Table 1. Table 2 comprehensively lists the possible semantic labels associated with each branching dialog state in the job interview application. The ultimate aim of the task is to provide interactive feedback to language learners about whether they have demonstrated the linguistic skills necessary to provide appropriate, intelligible responses to the interviewer's questions and to complete the communicative task successfully.

4 Model Architectures for SLU

Predicting semantic labels for spoken utterances in the job interview conversations is formulated as a problem of semantic utterance classification, which aims at classifying a given utterance into one of M semantic classes, $\{c_1^k, ..., c_M^k\}$, where k is the dialog state index and M is the total number of semantic labels defined for a given dialog state. A straightforward way to do semantic utterance classification is to use a sequence-to-tag function, which maps a sequence of input feature vectors, $O = \{o_1, o_2, ..., o_T\}$, to a semantic label, c^k. The conventional

[1] https://halef.org

Table 1 An example of different responses (along with corresponding gold-standard semantic labels) at one particular dialog state (Mistake) in the job interview task that deals with how the interviewee would deal with a co-worker's mistake.

System question	Imagine you saw your coworker make a mistake. Which do you think would be better? To tell the co-worker about the mistake or to speak with your manager?
Sample response 1	I would talk to the team member and ask him to rectify their mistake and it is a better way of resolving the issue.
Semantic label	coworker
Sample response 2	Speaking with the manager is the best thing I guess.
Semantic label	manager
Sample response 3	Yeah if it is a normal issue, then I'll go and discuss with the uh uh coworker himself. If it is something big, then I'll go to manager and I will discuss with him and we will come to the solution.
Semantic label	depends
Sample response 4	Uh uh currently I am staying in India. Eh.
Semantic label	nomatch

conversion from speech signals to semantic labels contains separate models for the different stages, as follows:

$$\hat{c^k} = \underset{c^k}{\operatorname{argmax}} P(c^k|W,\theta^c)P(W|\theta^w)P(W|H,\theta^h)P(H|O,\theta^o) \quad (1)$$

where $W = \{w_1, w_2, ..., w_N\}$ is a word sequence; $H = \{h_1, h_2, ..., h_J\}$ is a phone sequence; θ^c, θ^w, θ^h and θ^o are the parameters of the NLU model, the LM, the pronunciation dictionary, and the AM, respectively. This formulation assumes that these four models are conditionally independent of each other; accordingly, different corpora can be used to train each model. Generally, more than a hundred hours of speech collected under real usage conditions (along with associated transcriptions and semantic labels for acoustic, language, and NLU modeling or adaptation) are required to achieve reasonable SLU performance. When using deep learning methods, it is important to consider that the speech recognition performance typically increases monotonically with more training data [31]; thus, any new application can be continuously improved by using an iterative cycle of data collection and model refinement. The performance of the NLU and the LM can be further enhanced by adding corpora that contain only text from the same domain into the training set.

As mentioned in Section 1, the performance of SLU decoding in the context of a conversational language learning application can suffer a variety of challenges, ranging from a multitude of different acoustic environments, accented pronunciations, language model mismatch, and ASR errors. To break down the different issues leading to degraded SLU performance, we explore modeling SLU from the acoustic, phonetic, lexical, and semantic spaces, i.e., learning the sequence-to-tag function using BLSTM RNN models with acoustic feature sequences, phone posteriorgrams, and ASR word hypotheses as the input and concatenating the three BLSTM-RNNs together to compensate for the loss of information in different models. The schematic diagram of our approach to SLU is shown in Fig. 1.

RNNs configured to process arbitrary length input sequences have been successfully applied to solve a wide range of machine learning problems with sequence data. With BLSTM cells [32], an RNN can overcome the vanishing gradient problem in training. For a sequence-to-tag function such as semantic utterance classification for spoken dialog systems, the output layer of the BLSTM-RNN is a softmax layer which contains semantic labels represented by a one-hot vector and the input layer contains feature vectors along the time axis. The semantic label posteriors generated from three BLSTM-RNNs are concatenated together and modeled by a Support Vector Classifier (SVC) to predict the semantic labels again as final predictions, which can be regarded as a score level fusion process. An alternative fusion method can be feature level fusion, in which the output generated from the middle layers

Table 2 Dialog state and semantic labels.

Dialog State	Semantic Labels
Mistake (MT)	coworker, depends, manager, nomatch
Part or Full (PF)	either, full-time, nomatch, part-time
Self or Group (SG)	both, group, nomatch, self
Work Experience (WE)	yes, no, nomatch

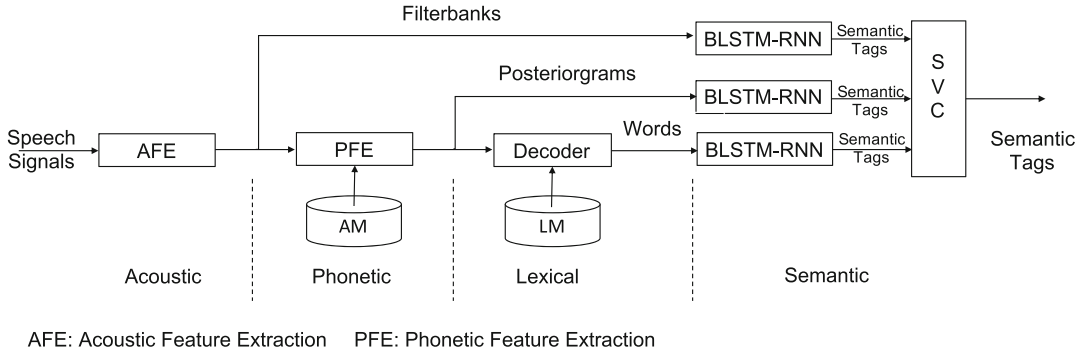

Figure 1 A schematic diagram of SLU with acoustic features, sub-phone posteriorgrams and ASR word hypotheses.

(instead of the output layers) of the three BLSTM-RNNs are concatenated together to predict the semantic labels.

4.1 ASR-free End-to-end modeling (E2E) for SLU

We propose to use an end-to-end modeling approach to directly model the relations between the given acoustic feature sequence and the corresponding semantic label, thereby obviating the need for the NLU model, LM, pronunciation dictionary, and AM from Eq. 1, as follows:

$$\hat{c^k} = \underset{c^k}{\operatorname{argmax}} P(c^k|O,\theta) \tag{2}$$

where θ is a set of parameters of the end-to-end model. We initially tried to use frame-level spectral information as input for predicting semantic labels with a BLSTM-RNN model directly; however, the preliminary results were not encouraging. We conjecture that the approach might be constrained by the limited training data used in our experiments and the resultant model either overfits the training data or is mismatched with the testing data. Speech acoustic features can vary greatly due to a variety of factors, e.g., age, gender, accent, and personalized speaking style. Therefore, a large number of spoken utterances tagged with corresponding semantic labels would likely be required to achieve reasonable classifier performance.

Therefore, we propose to use a compact representation for the utterance in variable length and then employ the resultant low-dimensional feature vector for semantic label modeling. We employed a pyramid BLSTM-RNN structure, as proposed in [33] and presented in Fig. 2. The pyramid structure makes the model training converge quickly. The final encoder layer is a fixed-dimensional vector V, which can be regarded as spoken sentence embedding. A resampling strategy [33] is used in the training of the pyramid structure to reduce the likelihood of overfitting. The encoder layers are initialized (pre-trained) by an RNN-based acoustic autoencoder [34, 35] in which the acoustic feature vector sequence is mapped onto a fixed-dimensional vector with the encoder RNN, and the

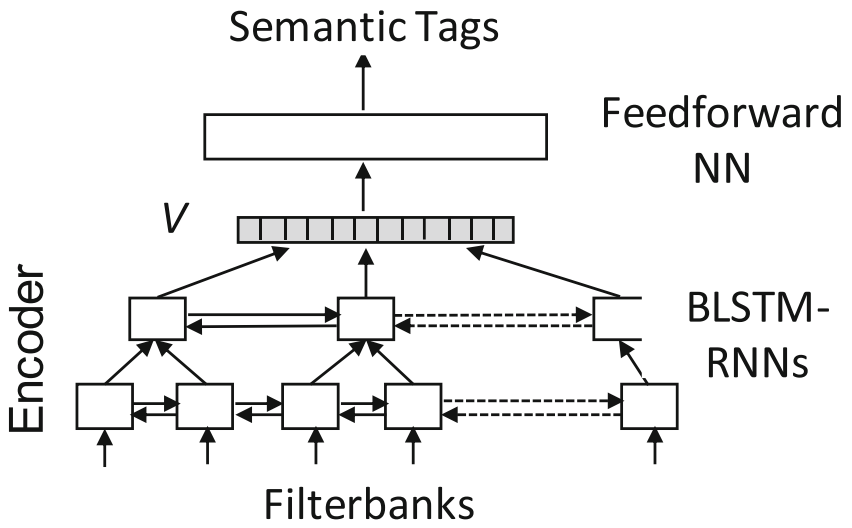

Figure 2 ASR-free End-to-end modeling to SLU.

decoder RNN reconstructs another sequence from the fixed-dimensional vector to minimize the reconstruction error. It is a feature compression based approach with unsupervised learning. Additional speech data without transcriptions can be used to enhance the pre-training performance.

Transfer learning or multi-task learning [36] can exploit commonalities between the training data of different learning tasks so as to transfer learned knowledge from one to another. We use multi-task learning for semantic utterance classification by treating each dialog state as a separate task. The schematic diagram of our approach is shown in Fig. 3 where the input layer is the fixed-dimensional vector V output from the BLSTM-RNN encoder as the representation of a variable-length acoustic feature vector sequence and the output layer is the softmax layer with K one-hot vectors (each vector represents one dialog state).

4.2 Subphone/Phone Posteriorgram for SLU

The user's spoken response to the conversation system is produced as spontaneous speech, which means that the word sequence is often difficult to predict if we have to use an LM that is trained on a corpus of written text or transcriptions of read/prepared speech due to insufficient training data. Spontaneous speech produced by non-native speakers is even harder to predict due to the presence of grammatical and vocabulary errors as well as non-standard uses of words and phrases. A subphone or phone lattice, a compact representation of the LVCSR phonetic search space, was employed for the task of audio information retrieval [37] to address the issues of the high perplexity of the LM and the presence of OOV words in queries and the corpus. Inspired by this approach, we explore SLU modeling using subphone/phone inputs by skipping the LM. The semantic labels can be predicted as follows:

$$\hat{c^k} = \underset{c^k}{\operatorname{argmax}} P(c^k|H,\theta)P(H|O,\theta^o) \tag{3}$$

DNN-based acoustic models for LVCSR use senones (which are tied tri-phone states from the HMM) as the output nodes of the DNN. The senones and the corresponding aligned speech frames from the GMM-HMM are used to train the DNN. In decoding, given a frame-level feature vector, the senone posteriors are generated as the DNN output. A subphone (senone) posteriorgram is the posterior distribution across the whole senone set over all frames in an utterance. A phone posteriorgram, i.e., the posteriors of the phones over time, is generated by summing up the posteriors of senones of the same phone. It is a matrix where the horizontal axis is time or the frame index, while the vertical axis is marked with subphone/phone indices, and the cell value is the posterior of the subphone/phone h at time t.

A BLSTM-RNN model is employed for SLU with the subphone/phone posteriorgram, as shown in Fig. 4. Given an input vector sequence $x_t \in \{x_1, ..., x_T\}$ where x_t is t-th frame vector containing the posterior probabilities of subphones or phones and T is the total number of frames in an utterance, it outputs a vector $C = c_1, ..., c_M$ indicating the posteriors of the semantic labels, i.e., the average of the last state of the forward state sequence, y_T^f , and the first state of the backward state sequence, y_1^b:

$$h_t = \mathcal{H}_{\mathcal{LSTM}}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{4}$$

$$y_t = W_{hy}h_t + b_y \tag{5}$$

$$C = mean(y_T^f, y_1^b) \tag{6}$$

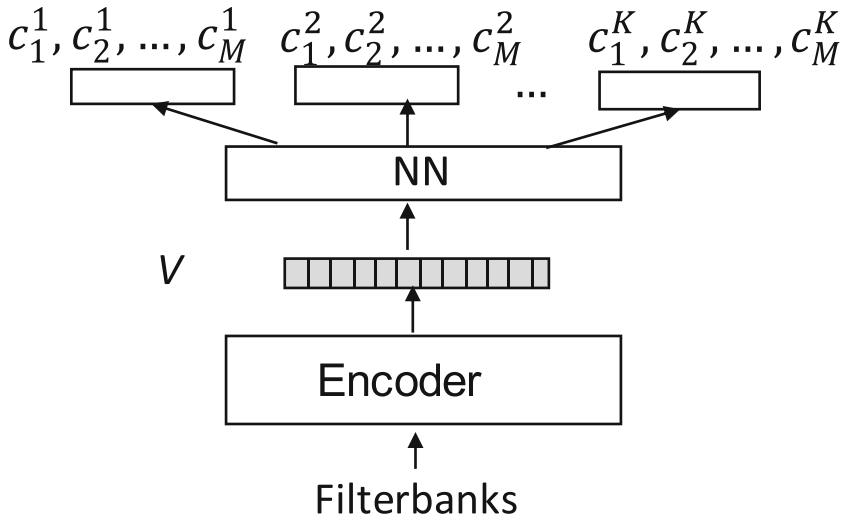

Figure 3 Transfer learning with feedforward NN.

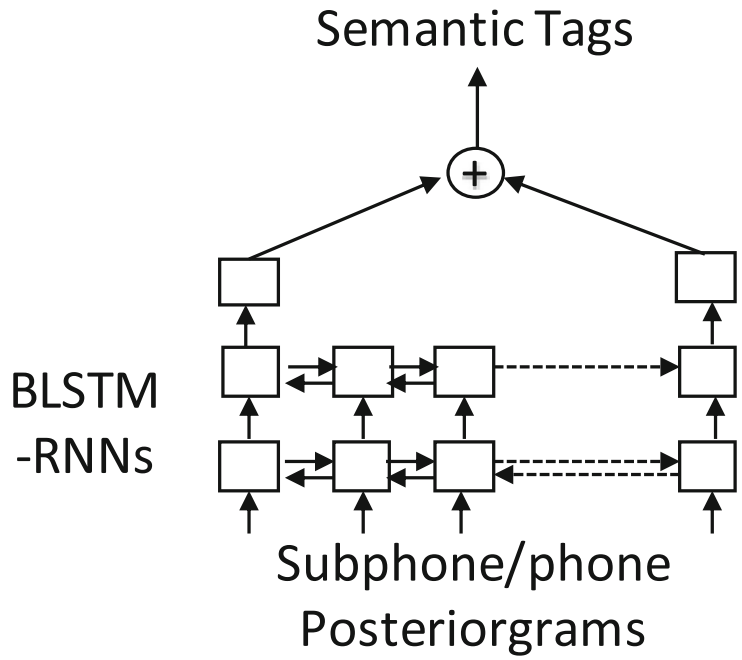

Figure 4 Subphone/phone Posteriorgram for SLU.

4.3 Word Embedding for SLU

Conventional statistics-based approaches to NLU use a bag-of-words representation based on the overall frequency of occurrence of each word to train a semantic utterance classifier. However, this approach does not adequately capture the full nature of the speech communication process since it omits contextual information and temporal dynamics. Therefore, we employ a BLSTM-RNN model for SLU with word embeddings, as presented in Fig. 5, in which each recognized word is represented by a vector from a Word2Vec model as input to the BLSTM-RNN. ASR recognition results, in terms of a sequence of recognized words, traverse the embedding layer and the BLSTM-RNN layer. In the embedding layer, the word $w_t \in \{w_1, ..., w_N\}$ represented by its *one-hot* representation is projected into a d_E dimensional space, e_t:

$$e_t = \mathcal{H}_E(Ew_t) \tag{7}$$

where E is the word embedding matrix initialized by Google's Word2Vec model and optimized during model training. The BLSTM-RNN layer and the output softmax

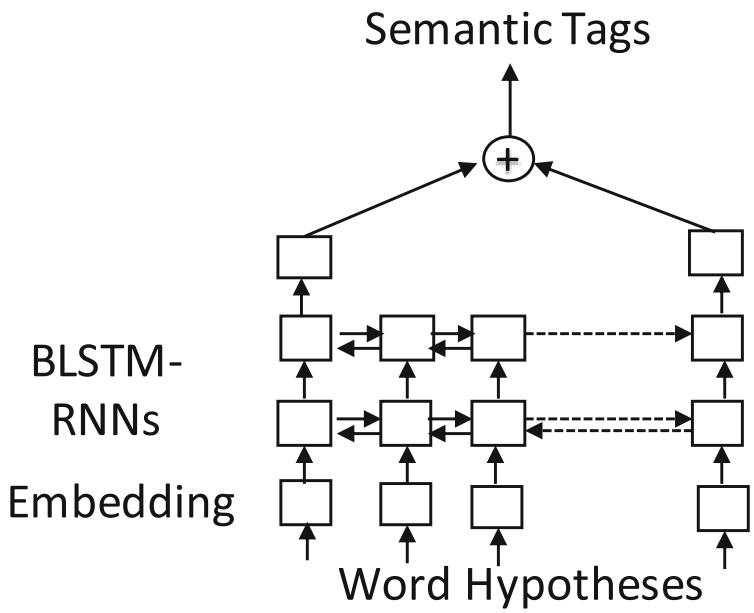

Figure 5 Word Embedding for SLU.

layer are the same as in Fig. 4 for SLU with the subphone/phone posteriorgram.

5 Experiments and Results

The BLSTM-RNN-based approaches to semantic utterance classification with the input features extracted in acoustic, phonetic, lexical spaces and the fusion of these three classifiers are evaluated in a spoken-dialog-based language learning application for non-native speakers, as described above.

5.1 Corpora

We collected spoken dialog data via the Amazon Mechanical Turk platform. Crowdsourced non-native speakers of English interacted with the spoken dialog system in the job interview task described above. Participants were compensated approximately $1.50 for completing the task, which consisted of reading the instructions, conducting the conversation, and completing a post-task survey; the average completion time for the entire task was 10 minutes. For the SLU experiments, we extracted all of the responses to the four branching dialog states listed in Table 2; the resulting corpus consists of 4,776 utterances spoken by 1,179 speakers. 4,191 utterances are used as a training set and the remaining 586 utterances are used as a testing set. 200 utterances randomly selected from the corpus were used to manually check the audio quality by having annotators view the waveform and spectrogram together while listening to the audio. Based on these annotations, we found that the percentage of labels for bad quality (i.e., perceptibe clipping distortion, packet loss, or substantial background noise), no voice, and good quality are 62.5%, 8.5% and 29%, respectively. This distribution of labels on this subset of the corpus demonstrates how challenging this data set is due to the poor audio quality in the majority of the responses. The quality of transcriptions was also checked by computing the Levenshtein distance between transcriptions from different independent transcribers for the same utterance, i.e., calculating the word error rate by assuming that one transcription is the reference and the second one is a recognition hypothesis. This analysis again demonstrates the challenging nature of this data set: the average inter-transcriber WER measured on 1,004 utterances / 10,288 tokens was 38.3%. This corpus is hereafter referred to as the job interview task (JIT) corpus.

Two additional corpora were used to build the ASR system. One is drawn from a large-scale global assessment of English proficiency that measures a non-native speaker's ability to use and understand English at the university level. The speaking tasks in this assessment elicit monologues of 45 or 60 seconds in duration; example tasks include expressing an opinion on a familiar topic and summarizing information presented in a lecture. This corpus contains over 800 hours of non-native spontaneous speech covering over 100 L1s (native languages) across 8,700 speakers and is hereafter referred to as the non-native speech (NNS) corpus.

Another corpus was collected using the HALEF SDS via crowdsourcing for a range of conversation-based language learning applications (including the job interview task, but with no data overlapping with the JIT corpus). This corpus is collected under realistic usage conditions; the acoustic environments and speaking styles match those in the JIT corpus. This corpus contains 41,185 utterances (roughly 50 hours) draw from five language learning tasks (about 37% of the responses from the job interview task) and is hereafter referred to as the SDS corpus.

5.2 ASR System

ASR systems were constructed using the Kaldi toolkit [38]. A GMM-HMM was first trained to obtain senones (tied tri-phone states) and the corresponding aligned frames for DNN training. The input feature vectors used to train the GMM-HMM contain 13-dimensional MFCCs and their first and second derivatives. Context-dependent phones, triphones, were modeled by 3-state HMMs and the pdf of each state was represented by a mixture of 8 Gaussian components. The splices of 9 frames (4 on each side of the current frame) were projected down to 40-dimensional vectors by linear discriminant analysis (LDA), together with maximum likelihood linear transform (MLLT), and then used to train the GMM-HMM by using maximum likelihood estimation. Concatenated MFCC features and i-vector features were used for DNN training. The input features stacked over a 15 frame window (7 frames to either side of the center frame for which the prediction is made) were used as the input layer for the DNN. The output layer of the DNN consists of the senones of the HMM obtained by decision-tree based clustering. The input and output feature pairs were obtained by frame alignment for senones with the GMM-HMM. The DNN has 5 hidden layers, and each layer contains 1,024 nodes. The sigmoid activation function is used for all hidden layers. All the parameters of the DNN were first initialized by pre-training, then trained by optimizing the cross-entropy function through back-propagation (BP), and finally refined by sequence-discriminative training, state-level minimum Bayes risk (sMBR).

5.3 BLSTM-RNN Configurations

BLSTM-RNNs with acoustic features, subphone/phone posteriorgrams or ASR hypotheses as input features and semantic labels as output nodes are constructed using

the Keras Python package[2]. 15% of the training data is randomly selected to tune the parameters of the BLSTM-RNN and avoid overfitting by using early stopping. The structures of the BLSTM-RNNs are configured for the three different sets of input features as follows.

5.3.1 Acoustic Features

The input acoustic features to the BLSTM-RNN are 26-dimensional Mel-frequency filterbanks (computed with a 25 msec. window, shifted every 10 msec.) without delta features or stacked frame window since the RNN architecture already captures long-term temporal dependencies among all sequential events. Silence segments at the beginning and end of utterances are deleted with an energy-based voice activity detection (VAD) module.

A two-layer BLSTM with 256 nodes for the first layer and 128 nodes for the second layer is employed. A layer with 400 nodes is used to compute the embedding from encoder layers. We unfolded encoder RNNs for 10 seconds or 1,000 time steps (frames) where 10 seconds is the median length of utterances in our corpus. Depending upon the length of the utterance, features are either padded with zeros at the end or down-sampled to 1,000 frames. A back-propagation through time (BPTT) learning algorithm is used to train the BLSTM-RNN parameters. A 400-dim embedding vector is then fed into a feed-forward NN with two hidden layers (each layer with 128 nodes) to predict semantic labels. All parameters of the NN are trained by optimizing the cross-entropy function through BP.

Two hidden layers, each layer with 128 nodes, are used for multitask learning with a feedforward NN. The input layer of the NN is the 400-dimensional V and the output layer of the NN has 15 nodes separated by four tasks. All parameters of the NN are trained by optimizing cross-entropy function through BP. The parameters in the hidden layers are updated by using all data in the training set of the JIT corpus while the corresponding dialog state dependent data is used to update the parameters in the top layer of the NN.

5.3.2 Subphone/Phone Posteriorgrams

A subphone/phone posteriorgram of an utterance is a time sequence of vectors, or equivalently, a 2D tensor (with a shape of # frames $\times$ # subphones/phones). We construct a tensor with a 100 $\times$ 3,686 shape for subphones (senones) or a 300 $\times$ 348 shape for phones (word-position-dependent phones) by resampling each spoken utterance into a fixed number of frames, i.e., 100 for subphones and 300 for phones, as the input for BLSTM-RNN training. The structure of the BLSTM-RNN is configured as 32 LSTM cells, a rectified linear unit (ReLU) activation function and a one-half drop-out rate (p=0.5); a categorical cross-entropy loss function and Adadelta optimizer is used in training.

5.3.3 ASR Hypotheses

The input ASR hypothesis sequence is similarly converted to a 2D tensor which is fed into a stacked BLSTM-RNN and then formalized as a vector to predict the semantic labels by the softmax output layer. The structure of the BLSTM-RNN is the same as that of the subphone/phone posteriorgram except that the input is a tensor with a shape (50 $\times$ 300), in which the maximum number of recognized words in an utterance is 50 and the dimension of word embedding vectors is 300, as trained from the Google News corpus[3].

We also use a bag of words model as a feature for training the semantic utterance classifier. In this model, a text string (the ASR recognition hypothesis) is represented as a vector based on the frequency of occurrence of each word. Dialog state-dependent models are trained to perform multi-class classification using bag of words features.

5.4 BLSTM-RNN Fusion

We tried a range of different classification models including Support Vector Machine (SVM), Random Forest, Logistic Regression, AdaBoost Decision Trees, etc., that are provided in the SKLL toolkit[4] to perform score-level fusion. In this process, the semantic label posteriors generated from each of the three BLSTM-RNN models are used as the input to a classifier to predict a final semantic label for the utterance. The hyperparameters of these classifiers were optimized by SKLL internally using cross-validation on the training data. We also tried a more straightforward approach in which the three BLSTM-RNNs were concatenated and MLP layers were added on top to predict the final semantic labels. In addition, we also explored feature-level fusion and a single, hierarchical BLSTM-RNN using all of the input types as comparison approaches.

5.5 Experimental Results

The performance of different ASR systems, in terms of WER, on the test set from the JIT corpus is shown in Table 3. The WERs are broken down by dialog state as well as those of overall (All) and the reference (Ref), which are tested on the matched data sets. The state-of-the-art DNN-based ASR trained on the Fisher corpus [39] using Kaldi can achieve 22.2% WER on its own testing set [40]. Although

[2] https://keras.io

[3] https://code.google.com/archive/p/word2vec

[4] https://github.com/EducationalTestingService/skll

Table 3 WER(%) by dialog state of ASR systems built with different corpora.

Dialog State	PF	WE	SG	MT	All	Ref
Fisher	86.4	88.8	84.2	90.9	88.1	22.2
NNS	54.3	62.6	52.0	54.8	55.5	18.5
SDS	35.4	55.8	45.0	55.1	49.4	N/A
NNS+SDS	35.8	50.1	39.5	46.1	43.5	N/A

the Fisher corpus is a collection of conversational telephone speech, it still exhibits a significant mismatch to the speech collected in the SDS corpus and results in a very high WER. The DNN-based ASR system with i-Vector based speaker adaptation technology trained on the NNS corpus (which is also a collection of non-native speakers' speech), can obtain WERs of 18.5% and 23.3% on monologic and dialogic data sets, respectively, using LM interpolation technology to compensate for the speaking style difference across tasks [41]. However, when it is applied to recognize the data collected from the SDS application, the WER is degraded to 55.5% even when transcriptions from the training set of the JIT corpus are used for language model adaptation. Using data collected by the SDS application or combining the NNS corpus with the SDS corpus can significantly improve the ASR performance: the WERs on the JIT test set are reduced to 49.4% and 43.5%, respectively. While this is still a very high WER value, it should be considered in light of the fact that the average inter-transcriber WER is also quite large at 38.3%, as reported above. Reducing both the ASR and inter-transcriber WERs for such data will be important steps for improving system performance in real-world environmental conditions and use cases, and pose an interesting challenge to the speech processing research community going forward.

Table 4 shows the SLU performance in terms of semantic prediction accuracy obtained by different SLU systems. The ASR-free E2E system, i.e., the BLSTM-RNN with acoustic features as input, performs much better than the majority vote baseline with an improvement from 59.8% to 70.6%, and there is no performance degradation for any of the dialog states. These results demonstrate that an ASR-free SLU is promising in situations with low ASR accuracy. We conjecture that the performance of an ASR-free SLU will be further improved if more training data is available. The acoustic features extracted from the NNS and SDS corpora are used to train the BLSTM-RNN auto-encoder in the sense of unsupervised learning. The SDS and NNS corpora can cover a large amount of acoustic variations and the V extractor trained on them with pre-training can yield superior discrimination for semantic classification. The overall accuracy of E2E(Pre) is improved by 3.5%, compared with that of E2E(JIT), where the V extractor is trained on the JIT corpus; the PF (Part or Full) dialog state achieves the largest gain among the four dialog states and shows an accuracy improvement of 6.9%. The multi-task learning approach can learn the commonalities among the different dialog states and further boost the overall accuracy (All) of semantic labeling from 68.9% to 70.6%.

Table 4 Accuracy(%) of different SLU systems.

Dialog State	PF	WE	SG	MT	All
Majority Vote	53.6	79.4	45.7	70.3	59.8
E2E(JIT)	57.4	80.3	54.2	76.1	65.4
E2E(Pre)	64.3	83.0	55.7	77.5	68.9
E2E(Pre,Multi-task)(a)	66.3	83.3	62.2	76.8	70.6
Subphone(b)	88.4	88.2	70.7	82.6	82.1
Phone	86.5	86.7	72.4	83.5	81.9
RF(BOW)	85.2	89.3	68.7	63.4	78.5
BLSTM-RNN(WE)(c)	89.5	89.2	77.4	85.5	85.1
Fusion(a+c)	90.6	90.2	78.0	85.5	85.8
Fusion(a+b+c)	90.6	93.1	79.3	87.0	87.0

Using the subphone posteriorgrams instead of acoustic features as the input to the BLSTM-RNN results in an improvement in semantic labeling accuracy (All) of 11.5%, from 70.6% to 82.1%, when compared with the E2E(Pre, Multi-task). The BLSTM-RNN using phone posteriorgrams as input can achieve similar semantic labeling labeling but with a much smaller (3.5 times smaller) input matrix. Phone posteriorgrams can be regarded as the posterior trajectories of phones in the phone set and represent a compact search space for the final speech recognition outputs with the LM. It has been shown that a phonetic lattice search can compensate for some of the information lost due to OOV words and improve the efficiency of information retrieval from spoken documents [43] and keyword spotting in large speech databases [42]. SLU in a certain sense can be considered as a special case of information retrieval or keyword spotting and a similar phenomena is possibly observed for SLU with the subphone/phone posteriorgram, which is equivalent to a subphone/phone lattice with posteriors assigned to each subphone/phone.

The conventional SLU approach, i.e., using ASR hypotheses as input for predicting the semantic label, can achieve 78.5% accuracy by using a Random Forest (RF) classifier a with Bag-of-Words (BOW) model, and 85.1% accuracy by using a BLSTM-RNN model with word embeddings (WE). The RF classifier with the BOW input achieves the best performance on semantic labeling among all the classifiers provided by SKLL. We think the benefits of using a BLSTM-RNN model with word embeddings for SLU come from two aspects: a) semantically similar words have similar word embedding vectors while the bag of words uses the same representation for the semantically

similar and dissimilar words; b) the BLSTM-RNN can capture long contextual information in both the forward and backward directions while the BOW model cannot tell the position of the word in a sentence, i.e., it cannot differentiate between sentences that contain the same words but in different orders.

The performance of different fusion methods is listed in Table 5. All fusion methods can improve the performance in terms of the accuracy of predicted semantic labels comparing with the performance of BLMSTM models with a single input type. The accuracy achieved by different fusion methods ranges from 86.2% to 87.0%, i.e., the performance of our employed fusion methods is not significantly different for this task. Among all these classifiers for the score-level fusion, the SVM classifier achieved the highest accuracy for semantic label prediction. The MLP method, i.e., concatenating three BLSTM-RNNs as MLP inputs to predict the final semantic labels, slightly under-performed the SVM classifier. Table 5 also shows that neither of feature-level fusions (a single or hierarchical BLSTM-RNN using all of the input types) outperformed the score-level fusion model with the SVM classifier. The classifiers provided by the SKLL toolkit train the fusion model by using all training data and optimize the hyperparameters using cross-validation while the neural network-based fusion methods have to separate 15% of the training data as a development set for tuning the hyperparameters due to their characteristics. We conjecture that the slightly inferior performance achieved by the neural network-based approaches is caused by the factor that the parameters in the trained model for fusion do not have full coverage of the statistics in the training data set.

The best fusion result in Table 5 is added into Table 4 as a comparison to those results without fusion. It indicates that acoustic features, subphone posteriorgrams and ASR word hypotheses have non-overlapping information and can compensate for each other in semantic tagging. The score-level fusion by using the posteriors output from the three RNNs can further improve the semantic labelling accuracy from 85.1% to 87.0%. It is enlightening to find that the performance can be improved by different combinations of BLSTM-RNNs. For example, both fused systems, i.e., filterbanks and ASR words (a+c) and filterbanks, posteriorgram and ASR words (a+b+c), can achieve a better performance than that of each individual system.

Table 5 Accuracy(%) of fusion with different methods.

Fusion	Score-level	Feature-level
Support vector machine	87.0	
Random forest	86.8	
Logistic regression	86.4	
AdaBoost decision tree	86.5	
MLP	86.7	
Single BLSTM-RNN		86.2
hierarchical BLSTM-RNN		86.6

5.6 Analysis and Discussion

The best performance of the BLSTM-RNN model using word embeddings is only 3.0% better than that predicted from subphone posteriorgrams (85.1% compared to 82.1%). In other words, the SLU model based on subphone posteriorgrams without using an LM trained on manual transcriptions for new application data can perform almost as well as the traditional ASR-based baseline in terms of semantic classification performance. We observed that the lower the ASR WER, the higher the accuracy of the SLU, as expected. When the ASR system was built only using the SDS corpus, the WER on the test set from the JIT corpus increases to 49.4% and the corresponding accuracy of semantic labeling degrades to 81.3%. It is interesting to note that the SLU model trained on manual transcriptions does not outperform the SLU model trained on the hypotheses produced by ASR system. We suspect that this might be due to inconsistencies and ambiguities present in the human transcriptions (due to the low intelligibility of the speech).

One of motivations for using acoustic filterbank features and subphone posteriorgrams is to investigate their possible advantages for the case of OOV words which cannot be recognized by the ASR system. 131 out of 586 utterances in the test set contain OOV words. The average tagging accuracy for the utterances with/without OOVs is shown in Table 6. The model based on subphone posteriorgrams outperforms the model based on ASR hypotheses for OOV utterances, i.e., the tagging accuracy is improved from 78.2% to 80.4%. However, for the utterances without OOV tokens, the relative ranking of tagging accuracies is reversed, i.e., 87.1% for using ASR hypotheses vs 82.6% for using senone posteriors. The performance gap between the utterances with OOV tokens and without OOV tokens is much larger for using ASR hypotheses than for using filterbanks and posteriorgrams. The fusion results also show that tagging performance can be boosted more for the utterances with OOV tokens, from 78.2% (ASR hypotheses) to 83.0% (a+b+c fusion), than for the utterances without OOV tokens, where only a marginal improvement of 1.1%

Table 6 Performance (average tagging accuracy) of the utterances w/ and w/o OOV.

	(a)Filter banks	(b)Posterior gram	(c) ASR Words	Fusion a+b+c
with OOV	69.1	80.4	78.2	83.0
w/o OOV	71.0	82.6	87.1	88.2

is obtained, i.e., from 87.1% (ASR hypotheses) to 88.2% (a+b+c fusion). In a conversation-based dialog system eliciting spontaneous speech where OOV tokens can be common, the performance of the final SLU predictions can therefore be enhanced by incorporating the features from the acoustic and phonetic levels for semantic labeling.

6 Conclusions

In this paper, we have investigated the performance of spoken language understanding in a human-machine conversation-based language learning system. Three bi-directional LSTM-RNNs were employed for end-to-end modeling of the relationships between the utterance semantics and sequential information extracted from three different levels of granularity as follows: filterbanks at the acoustic level, posteriorgrams at the subphonemic level, and recognized words produced by the ASR system at the lexical level. The models were evaluated on a crowdsourced, spoken dialog speech corpus containing speech produced by non-native speakers of English in a job interview task. Experimental results show that the end-to-end modeling approach is particularly promising in situations with low ASR accuracy and that the accuracy of semantic labeling can be further improved with a score-level fusion approach combining the output from the three individual RNNs. In the future, we will use more speech data for testing the learning and interpolation capability of BLSTM-RNNs to further improve SLU performance.

References

1. Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, D.Y.G., Zweig, G. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *23*(3), 530–539.
2. Xu, P., & Sarikaya, R. (2013). Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *Proc. of ASRU* (pp. 78–83).
3. Tur, G., Hakkani-Tur, D., Heck, L., Parthasarathy, S. (2011). Sentence simplification for spoken language understanding. In *Proc. of ICASSP* (pp. 5628–5631).
4. Huang, Q., & Cox, S. (2006). Task-independent call-routing. *Speech Communication*, *48*(3), 374–389.
5. Gorin, A.L., Petrovska-Delacretaz, D., Riccardi, G., Wright, J.H. (1999). Learning spoken language without transcriptions. In *Proc. of ASRU* (Vol. 99).
6. Alshawi, H. (2003). Effective utterance classification with unsupervised phonotactic models. In *Proc. of NAACL HLT*, (Vol. 1 pp. 1–7).
7. Wang, Y.Y., Lee, J., Acero, A. (2006). Speech utterance classification model training without manual transcriptions. In *Proc. of ICASSP*, (Vol. 1 pp. 553–556).
8. Wang, Y., Skerry-Ryan, R.J., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. (2017). Tacotron: Towards end-to-end speech synthesis. In *Proc. Interspeech pp 4006–4010*.
9. Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R.A. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4779–4783). IEEE.
10. Sotelo, J., Mehri, S., Kumar, K., Santos, J.F., Kastner, K., Courville, A., Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. In *ICLR 2017 workshop*.
11. Heigold, G., Moreno, I., Bengio, S., Shazeer, N. (2016). End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5115–5119). IEEE.
12. Zhang, S.X., Chen, Z., Zhao, Y., Li, J., Gong, Y. (2016). End-to-end attention based text-dependent speaker verification. In: *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT 2016)* (pp. 171–178). IEEE.
13. Ubale, R., Qian, Y., Evanini, K. (2018). Exploring end-to-end attention-based neural networks for native language identification. In *Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 84–91). IEEE.
14. Geng, W., Wang, W., Zhao, Y., Cai, X., Xu, B. (2016). End-to-End Language Identification Using Attention-Based Recurrent Neural Networks. In: *Proc. INTERSPEECH* (pp. 2944–2948).
15. Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M.A., Schuller, B., Zafeiriou, S. (2016). Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5200–5204). IEEE. https://www.overleaf.com/project/5d41b311fb69574cddcacef7.
16. Audhkhasi, K., Rosenberg, A., Sethy, A., Ramabhadran, B., Kingsbury, B. (2017). End-to-end ASR-free keyword search from speech. *IEEE Journal of Selected Topics in Signal Processing*, *11*(8), 1351–1359. IEEE.
17. Lengerich, C., & Hannun, A. (2016). An end-to-end architecture for keyword spotting and voice activity detection. arXiv:1611.09405.
18. Li, B., Sainath, T.N., Sim, K.C., Bacchiani, M., Weinstein, E., Nguyen, P., Chen, Z., Wu, Y., Rao, K. (2018). Multi-dialect speech recognition with a single sequence-to-sequence model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4749–4753). IEEE.
19. Toshniwal, S., Sainath, T.N., Weiss, R.J., Li, B., Moreno, P., Weinstein E., Rao, K. (2018). Multilingual speech recognition with a single end-to-end model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4904–4908). IEEE.
20. Chan, W., Jaitly, N., Le, Q., Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4960–4964). IEEE.
21. Prabhavalkar, R., Rao, K., Sainath, T.N., Li, B., Johnson, L., Jaitly, N. (2017). A comparison of sequence-to-sequence models for speech recognition. In *Proc. Interspeech* (pp. 939–943).
22. Chiu, C.C., Sainath, T.N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R.J., Rao, K., Gonina, E., Jaitly, N. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774–4778) IEEE.
23. Prabhavalkar, R., Sainath, T.N., Wu, Y., Nguyen, P., Chen, Z., Chiu, C.C., Kannan, A. (2018). Minimum word error rate

training for attention-based sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4839–4843). IEEE.
24. Sainath, T.N., Chiu, C.C., Prabhavalkar, R., Kannan, A., Wu, Y., Nguyen, P., Chen, Z. (2018). Improving the performance of online neural transducer models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5864–5868). IEEE.
25. Qian, Y., Ubale, R., Ramanaryanan, V., Lange, P., Suendermann-Oeft, D., Evanini, K., Tsuprun, E. (2017). Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 569–576). IEEE.
26. Qian, Y., Ubale, R., Lange, P., Evanini, K., Soong, F. (2018). From speech signals to semantics - tagging performance at acoustic, phonetic and word levels. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE.
27. Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B., Bengio, Y. (2018). Towards end-to-end spoken language understanding. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5754–5758). IEEE.
28. Haghani, P., Narayanan, A., Bacchiani, M., Chuang, G., Gaur, N., Moreno, P., Prabhavalkar, R., Qu, Z., Waters, A. (2018). From Audio to Semantics:, Approaches to end-to-end spoken language understanding. arXiv:1809.09190.
29. Ramanarayanan, V., Suendermann-Oeft, D., Lange, P., Ivanov, A.V., Evanini, K., Yu, Z., Tsuprun, E., Qian, Y. (2016). Bootstrapping development of a Cloud-Based spoken dialog system in the educational domain from scratch using crowdsourced data. ETS Research Report Series, Wiley, https://doi.org/10.1002/ets2.12105.
30. Ramanarayanan, V., Suendermann-Oeft, D., Lange, P., Mundkowsky, R., Ivanov, A., Yu, Z., Qian, Y., Evanini, K. (2017). *Assembling the Jigsaw: How Multiple Open Standards Are Synergistically Combined in the HALEF Multimodal Dialog System. Multimodal Interaction with W3C Standards*, (pp. 295–310). Berlin: Springer.
31. Cheng, J., Chen, X., Metallinou, A. (2015). Deep neural network acoustic models for spoken assessment applications. *Speech Communication*, *73*, 14–27.
32. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
33. Chan, W., Jaitly, N., Le, Q.V., Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. of ICASSP*.
34. Audhkhasi, K., Rosenberg, A., Sethy, A., Ramabhadran, B., Kingsbury, B. (2017). End-to-end ASR-free keyword search from speech. In *Proc. of ICASSP*.
35. Chung, Y., Wu, C., Shen, C., Lee, H., Lee, L. (2016). Audio Word2Vec: unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. In *Proc. of Interspeech*.
36. Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.
37. Siohan, O., & Bacchiani, M. (2005). Fast vocabulary-independent audio search using path-based graph indexing. In *Proc. of Interspeech*.
38. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. (2011). The kaldi speech recognition toolkit. In *Proc. of ASRU*.
39. Cieri, J., Miller, D., Walker, K. (2004). The fisher corpus: a resource for the next generations of speech-to-text. In *LREC*, (Vol. 4 pp. 69–71).
40. Peddinti, V., Povey, D., Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. of INTERSPEECH* (pp. 3214–3218).
41. Qian, Y., Wang, X., Evanini, K., Suendermann- Oeft, D. (2016). Self-adaptive dnn for improving spoken language proficiency assessment. In *Proc. of Interspeech* (pp. 3122–3126).
42. Tetariy, E., Gishri, M., Har-Lev, B., Aharonson, V., Moyal, A. (2013). An efficient lattice-based phonetic search method for accelerating keyword spotting in large speech databases. *International Journal of Speech Technology*, *16*(2), 161–169.
43. Saraclar, M., & Sproat, R. (2004). Lattice-based search for spoken utterance retrieval. In *Proc. of ACL* (pp. 129–136).

Affiliations

Yao Qian[1] · Rutuja Ubale[1] · Patrick Lange[1] · Keelan Evanini[2] · Vikram Ramanarayanan[1,3] · Frank K. Soong[4]

Rutuja Ubale
rubale@ets.org

Patrick Lange
plange@ets.org

Keelan Evanini
kevanini@ets.org

Vikram Ramanarayanan
vramanarayanan@ets.org

Frank K. Soong
frankkps@microsoft.com

1 Educational Testing Service Research, San Francisco, CA, USA

2 Educational Testing Service Research, Princeton, NJ, USA

3 University of California, San Francisco, CA, USA

4 Microsoft Research Asia, Beijing, China