EXPLORING END-TO-END ATTENTION-BASED NEURAL NETWORKS FOR NATIVE LANGUAGE IDENTIFICATION

Rutuja Ubale, Yao Qian and Keelan Evanini

Educational Testing Service Research, USA

{rubale, yqian, kevanini}@ets.org

ABSTRACT

Automatic identification of speakers' native language (L1) based on their speech in a second language (L2) is a challenging research problem that can aid several spoken language technologies such as automatic speech recognition (ASR), speaker recognition, and voice biometrics in interactive voice applications. End-to-end learning, in which the features and the classification model are learned jointly in a single system, is an emerging field in the areas of speech recognition, speaker verification and spoken language understanding. In this paper, we present our study on attention-based end-to-end modeling for native language identification on a database of 11 different L1s. Using this methodology, we can determine the native language of the speaker directly from the raw acoustic features. Experimental results from our study show that our best end-to-end model can achieve promising results by capturing speech commonalities across L1s using an attention mechanism. In addition, fusion of proposed systems with the baseline system leads to significant performance improvements.

Index Terms— end-to-end learning, native language identification, attention mechanism, deep learning

1. INTRODUCTION

Native Language Identification (NLI) refers to the automatic process through which we determine the native language (L1) of an individual from their spoken or written samples in a second language (L2). Acoustic signals contain information about the speaker's identity such as age, gender, language background of the speaker, accent and emotional state. Identification of such cues from speech is valuable for improving the robustness of existing spoken language systems. The task of native language identification is similar to the more commonly studied tasks of language identification, accent classification, and dialect identification. Determining the L1 background of the speaker is a more challenging problem since we use the speaker's response in a second language. Speakers from a particular L1 group tend to show common characteristics in their speech, such as a distinct foreign accent, typical pronunciation errors, and patterns of intonation and duration. These commonalities in L2 production from speakers within L1 groups are typically due to L1 transfer effects from a lack of complete acquisition of the L2 and can form the basis for the task of speech-based NLI.

Determining a speaker's L1 can help improve the interaction between users and machines for many interactive voice applications aimed at computer assisted language learning (CALL). For example, information regarding a speaker's L1 can enable the system to provide feedback specific to the learner in a grammar or pronunciation error correction system or can help design a more personalized conversation with a dialog system targeted at improving L2 proficiency. Furthermore, automatic speech recognition (ASR) systems tend to show degradation in performance on accented/non-native speech. There has been some research on improving ASR performance by using dialect information to train dialect-specific ASR models [1] or incorporating dialect information into the model [2]. Similarly, ASR performance on L2 speech can be improved by training L1-specific acoustic models [3].

Features commonly used for text-based NLI from L2 learners' written samples generally include character-level *n*-grams, word and part-of-speech (POS) tags, syntactic dependencies, and spelling and grammatical errors. Statistical classifiers are then trained using this set of features on data labelled with corresponding L1 information. For speech-based NLI, acoustic features from the learners speech such as MFCC, phone-level confusion, and lexical features like language usage error are commonly used. Automatic NLI systems developed for the NLI shared task held at the BEA workshop at NAACL 2013 for determining L1 from written essays could achieve a performance of 84% while models developed for the Computational Paralinguistics Challenge (ComParE) [4] at INTERSPEECH 2016 were able to achieve around 81% accuracy [5] in determining L1 from spoken responses from 11 L1 backgrounds. Our previous efforts aimed at improving sub-phone TDNN based i-vector approach for NLI [6, 7] on the ComParE challenge corpus could further improve the performance to 88% accuracy compared to 81% accuracy of the best system from 2016 ComParE challenge.

Most of the successful approaches to NLI have relied on the use of probabilistic models such as a Gaussian mixture model universal background model (GMM-UBM) or i-vector framework as the front-end to factorize speech signals into speaker-related factors. This is followed by a back-end scoring model e.g., the cosine-similarity metric, linear discriminant analysis (LDA) and probabilistic linear discriminant analysis (PLDA). In the ComParE challenge, systems using i-vector based models all achieved approximately 70% or higher accuracy for identification of the 11 L1 languages [8, 9, 5]. The performance of spectrum-based approaches for the ComParE corpus was in the range 45%-58% accuracy [10, 11, 12], and was thus much inferior to the i-vector approaches. While i-vector approaches contain the learning of speaker features and back-end scoring as separate components, end-to-end frameworks allow us to perform both together in a single system. In this paper, we propose attention-based models for spectrum-based end-to-end NLI. We evaluate the effectiveness of our proposed approach on a corpus of 11 L1 languages. Our study shows that our proposed models can achieve reasonable good performance when compared to the i-vector system. Our findings show that the end-to-end networks learn complementary representations to i-vector systems and we get significant improvements by combining their scores.

2. RELATED WORK

End-to-end learning has become very popular for speech recognition in recent years. Traditional ASR systems that consist of an acoustic model, a pronunciation model and a language model require independent training for the three modules. End-to-end architectures allow all three modules to be learned jointly in a single module [2, 13, 14, 15, 16, 17, 18]. State-of-the-art speech recognition models that employ an attention network can show significant improvement [14, 16, 17, 19] in word error rates. Furthermore, end-to-end sequence-to-sequence networks have been applied successfully to text-to-speech synthesis [20, 21, 22] to convert character sequences to mel spectrogram which is then converted to speech. Recently, several studies [23, 24, 25] have attempted to replace the ASR and natural language understanding components for spoken language understanding in spoken dialog systems with a single end-to-end model that extracts the domain/intent directly from the users speech. End-to-end models have also been explored for speaker verification task [26, 27] and are seen to outperform conventional i-vector/PLDA systems.

Most of the work on NLI has focused on i-vector based systems. The best performing model for the Native Language ComParE 2016 challenge and improved systems built thereafter have focused on i-vector frameworks. Very few research studies have targeted the use of spectral features such as MFCC or filter bank features. While spectrum-based deep learning approaches outperformed the baseline system in the ComParE challenge, their performance was much worse than the i-vector based systems. Our research study attempts to improve the performance of spectrum based end-to-end models by taking inspiration from state-of-the-art speech recognition models.

3. END-TO-END NATIVE LANGUAGE IDENTIFICATION

We take inspiration from sophisticated end-to-end ASR models and extend it to the task of learning the native language ID. End-to-end NLI is similar to an audio classification task that takes a sequence of acoustic features and maps the feature vector into one of the L1 classes $c \in (c_1, c_2, ..., c_n)$. Our network uses log-Mel filter bank features, $x = (x_1, x_2, ..., x_t)$ as input and outputs a vector of posterior probabilities for each L1 class. The class with the highest prediction probability is selected as the recognized L1 of the speaker.

3.1. Listen, Attend and Identify

Recurrent Neural Networks (RNNs) which process sequences of inputs and model dependencies over time have been widely used in speech systems. Speech signals can contain thousands of frames and sequential processing of over thousands of time steps can be computationally expensive. To solve this issue, our end-to-end model is inspired from the state-of-the-art end-to-end ASR model, Listen, Attend and Spell (LAS) [14].

Similar to the LAS model, our model consists of two components: the listener and identifier. The listener is a three-layer Bidirectional Gated Recurrent Unit (Bi-GRU) encoder network that takes log-Mel filter bank features as input and transforms the input feature vector into a high-order feature representation. Layers of Bi-GRUs are stacked in a pyramidal structure. In every pyramidal Bi-GRU layer (pBGRU), the number of time steps t for the input signal x is reduced by one half. The listener encoder maps the input x to a high-level representation $h^{enc} = (h_1, h_2, ..., h_T)$ where T is the length of h^{enc} reduced from the input length t.

In each Bi-GRU layer the outputs of the forward and backward layers are concatenated together. The number of timesteps at the listener output are reduced to $t/8$ steps. The three-layer pBGRU architecture allows us to subsample from a large number of frames and reduce the dimensionality of the feature vector. The identifier is an attention-based classifier that applies attention to the output of the listener. The attention layer is followed by a fully connected feed-forward layer that generates posterior probabilities for each L1 class. Reducing the dimensionality of the feature vector enables us to apply attention to a shorter feature sequence making it easier to discriminate between the classes. We refer to this system as the Listen, Attend and Identify (LAI) system.

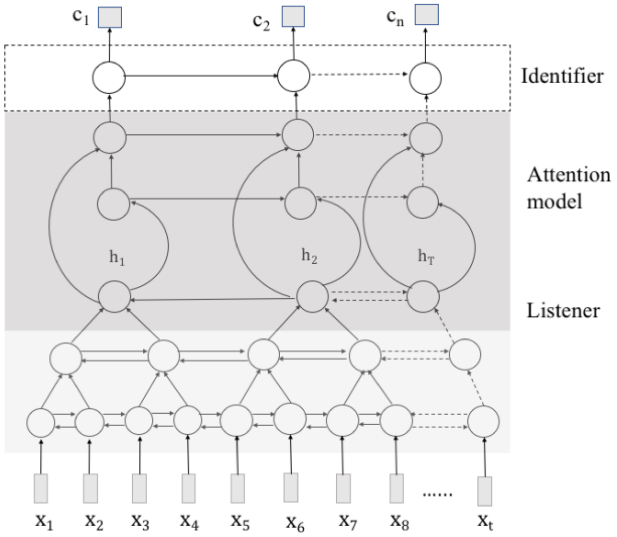


Fig. 1. Listen, Attend and Identify (LAI) architecture

3.2. CGDNN

We experimented with a second approach to subsampling long feature sequences by using a combination of Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU). The input signal $(x_1, x_2, ..., x_t)$ is represented with the dimensions $1 \times t \times k$, where t is the number of time steps and k is the length of the features at each time step. In our work, k is 40 since we use 40-dimensional log filter bank features. We have used four two-dimensional CNN layers to reduce the frequency variance in the input signal. Each CNN layer is followed by a max pooling layer where pooling is performed along the frequency axis. The output of the CNN network is a high-order feature representation of $feature\ maps \times time \times frequency$. To reduce the dimensionality of the feature vector further, we add a linear layer after the last CNN layer. The output of the linear layer is next passed to a network of uni-directional GRU layers that models the signal along the time domain. After performing temporal modeling, we apply attention to each time step in the output of the GRU network. Next, we pass the output of the attention layer to two fully connected DNN layers. We refer to this system as the CGDNN system, since it contains a combination of CNN, GRU, and fully connected DNN layers.

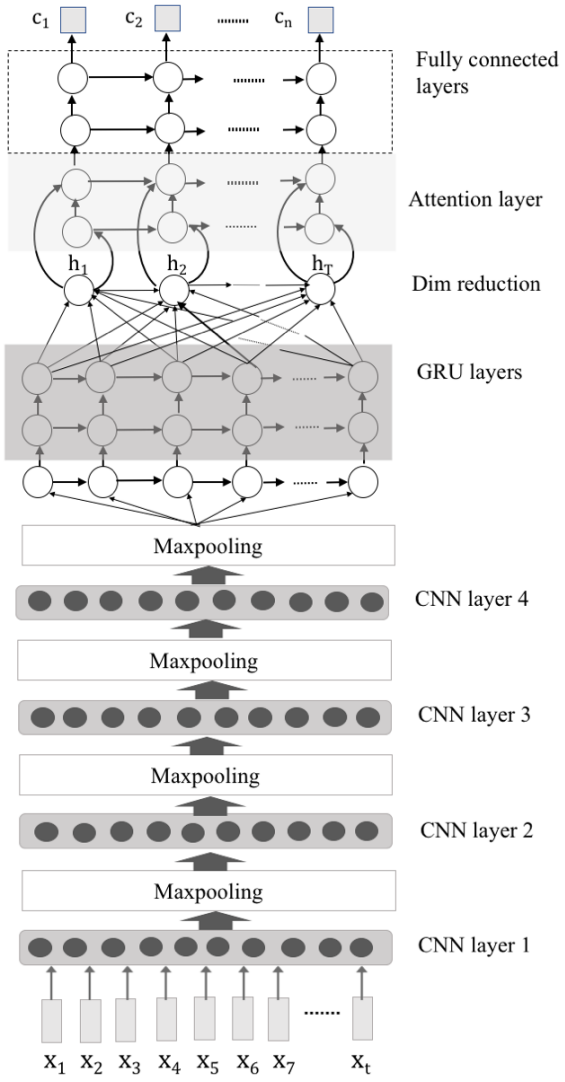


Fig. 2. CGDNN architecture

3.3. Attention mechanism

Recently, attention mechanisms are becoming quite popular and performing very well on many tasks such as speech recognition [14, 19], speaker verification [26, 27], machine translation [28], image captioning [29]. Our end-to-end architectures, LAI and CGDNN consist of three major components: an encoder network which maps input acoustic features to a high-level representation, an attention model that determines which parts in the feature representation to pay attention to and a fully connected classifier network that determines the probability of predicting all the classes given the feature vector from the attention model. Let $h^{enc} = (h_1, h_2, ..., h_T)$ be the high-level feature representation at the output of the acoustic encoder i.e. the output from the last GRU layer of both models defined in Sections 3.1 and 3.2, where T is the length of h^{enc}. The attention vector v is a representation of the important parts of the feature vector h^{enc}. We learn the score e_i for the encoder output h_i at each time step i using the equation:

$$e_i = \tanh(h_i), \quad i = 1, ..., T \tag{1}$$

The mechanism calculates a weight α_i for the encoder output for each time step *i*, which can be inferred as the probability of the importance of time step *i*. We use the softmax function to compute the normalized weight $\alpha_i \in [0, 1]$:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{k=1}^{T} \exp(e_k)} \tag{2}$$

where $\sum_{i=1}^{T} \alpha_i = 1$. The attention vector v is computed as the weighted average of the encoder outputs at all time steps:

$$v = \sum_{i=1}^{T} \alpha_i h_i \tag{3}$$

In addition to the basic attention mechanism mentioned above, we have also tried two variants of the attention layer [30]: cross-layer attention and divided-layer attention. For cross-layer attention, the attention vector is computed as shown in equation 3 as the weighted average of the final layer outputs of the encoder but with weights computed using the second-to-last layer. For LAI, we use the second-to-last pBGRU layer and for CGDNN, the second-to-last GRU layer is used to compute the scores e_i and attention weights α_i. For the second variant, we double the dimension of the last layer outputs and divide it equally into two parts. One part is used to compute the attention weights α_i and the other is used to compute the attention vector v.

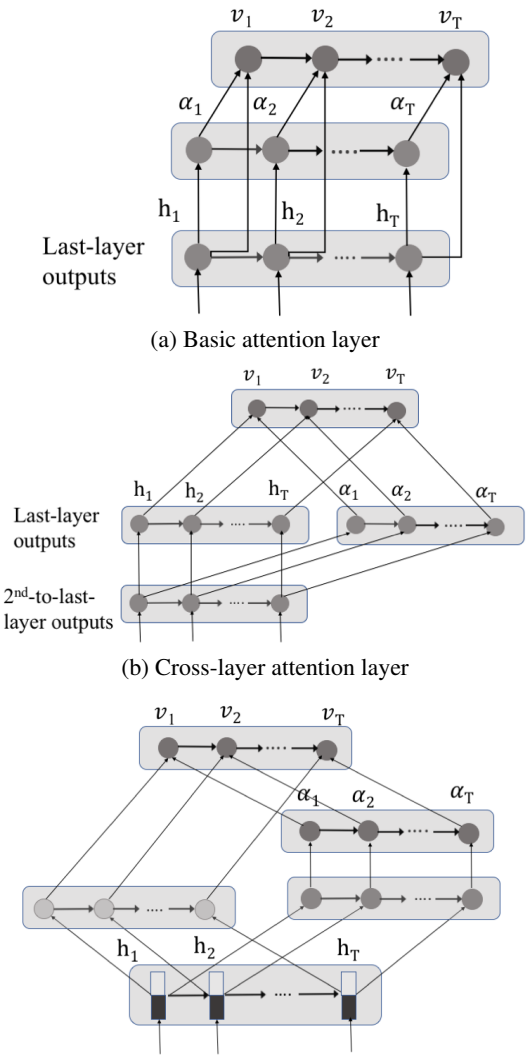
4. EXPERIMENTS

4.1. Corpora

Our experiments on end-to-end modeling and i-vector baseline for L1 recognition are evaluated on a corpus of non-native English speech collected during a high-stakes global assessment of English language proficiency. Our corpus is similar to the corpus provided by Educational Testing Service (ETS) as the Native Language sub-challenge corpus for ComParE Challenge at Interspeech 2016 [4]. The corpus consists of spoken responses from 11,000 non-native speakers with 11 different L1 backgrounds: Arabic (ARA), Chinese (CHI), French (FRE), German (GER), Hindi (HIN), Italian (ITA), Japanese (JAP), Korean (KOR), Spanish (SPA), Telugu (TEL) and Turkish (TUR). Each response is approximately 45-60 seconds long. The corpus contains approximately 138 hours of speech sampled at 16 kHz. There are approximately 1,000 speech recordings for each L1 in the dataset. The data is partitioned as follows with no overlapping speakers: we have used 7,040 recordings for training our models, 1,760 recordings are used as validation set and 2,200 recordings are used for testing the performance of our models. The test set contains 200 responses for each L1.

4.2. Experimental setup

The input features for all end-to-end models are 40-dimensional log-Mel filter bank features computed every 10 ms. The data is normalized to zero mean and unit standard deviation using mean and standard deviation from the training set. All end-to-end models are trained using Keras with a Tensorflow backend on three CUDA-enabled GPUs. The i-vector baseline system is trained using Kaldi. All end-to-end networks are trained using 50 epochs with a batch size of 32 samples.



(a) Basic attention layer

(b) Cross-layer attention layer

(c) Divided-layer attention layer

Fig. 3. Attention mechanisms: (a) basic attention (b) cross-layer attention (c) divided-layer attention

4.2.1. Listen, Attend and Identify (LAI)

In the Listen encoder, we have used three pBGRU layers with 512 nodes i.e. 256 nodes in each direction. Adding more layers to the network did not improve the performance. We also tried using pyramidal Bidirectional Long Short Term Memory (pBLSTM) layers instead of pBGRU layers. However, the accuracy achieved using pBGRU was slightly better than pBLSTM. Also since GRUs do not use memory unit like LSTMs and have less complex structure they are computationally more efficient; we saw a significant decrease in training time by using BGRUs over BLSTMs. The outputs of the forward and backward RNNs are concatenated together. For all pBGRU layers, the weights are initialized using the Glorot-Bengio initializer [4] and Hyperbolic Tangent (tanh) activation. To avoid overfitting, each pBGRU layer is followed by a dropout layer with a dropout rate of 30%. The Identifier consists of an attention layer and an output layer which is a fully connected feed-forward layer with softmax activation with a hidden size of 11 corresponding to the 11 L1 classes. The model is trained using categorical cross-entropy criterion and optimized using Adam optimizer [31] with a learning rate of 0.0005.

4.2.2. CGDNN

In the multi-layer CNN network, the first layer is a 2D convolution layer with 16 output filters in the convolution, kernel size of 7×7. This is followed by max pooling layer of pooling size 6×6. The second 2D CNN layer has kernel size of 5×5 and 32 output filters. The remaining two CNN layers have kernel size of 3×3 and 32 output filters. All CNN layers use Rectified linear unit (ReLU) activation. The last three CNN layers are each followed a max pooling layer of pooling size 3×1. We have added Batch Normalization [32] layers between every connection in the multi-layer CNN network. Adding batch normalization layers helped speed up the convergence of this architecture.

Between the CNN and RNN networks, we have connected a linear layer with 128 outputs. The RNN network consists of 2 GRU layers each with 256 nodes. The last RNN layer is followed by an attention layer connected to a feed-forward layer with 32 nodes followed by a fully connected softmax layer with 11 output nodes. Adding more layers did not improve the performance of the architecture. For all CNN and RNN layers the weights are initialized using the Glorot-Bengio initializer.

The model is trained using categorical cross-entropy cost function and optimized using Stochastic Gradient Descent (SGD) optimizer with drop-based learning rate decay and momentum of 0.8. We have implemented the decay function such that the learning rate (lr) is dropped by one half every 10 epochs. We used an initial learning rate of 0.1 which is dropped by one half every 10 epochs.

$$lr = intial\ lr * drop^{floor(epoch/epoch\_drop)} \quad (4)$$

4.2.3. CNN and RNN models

For our preliminary experiments, we developed a simple 4 layer CNN network similar to the 4 layer CNN network from the CGDNN model. We report the results of applying attention mechanism to the 4 layer CNN network in Section 5. To analyze the effect of adding GRUs to the CNNs in CGDNN architecture, we also report the results of using only a 2 layer GRU network. Both the models are trained using SGD optimizer with drop-based learning rate decay.

4.3. Baseline system

A GMM-based i-vector system is used as the baseline NLI system in this study. Based upon factor analysis, an i-vector is a compact representation of a speech utterance in a low-dimensional subspace. The i-vector based approach has been successful in recognizing speaker and language identity and is well-suited to NLI. In the baseline NLI system, an energy-based voice activity detection (VAD) method is applied to detect non-speech segments within utterances. 20 dimensional MFCCs including c_0, extracted from the resultant speech segments via a 20ms Hamming window with a 10ms time shift. MFCCs are appended with their first and second derivatives. Utterance-based cepstral mean normalization was performed on the acoustic feature vectors. A GMM with 2,048 Gaussian Kernels and a full covariance matrix was trained as the Universal Background Model (UBM) by using the training set of the corpus mentioned in Section 4.1. The same training set was also used to train an i-vector extractor T-matrix, i.e., a low rank rectangular matrix called total variability, as well as Probabilistic Linear Discriminant Analysis (PLDA)

Table 1. The accuracy obtained by different NLI systems on the test set of 11 L1 corpus

Method	Accuracy (%)	UAR (%)
Majority vote baseline	9.00	8.26
RNN only	42.09	42.87
CNN only	60.45	61.13
CGDNN	69.18	69.66
LAI	70.45	70.87
i-vector baseline	79.72	81.59

projection matrices. We employ PLDA as a scoring method to L1 recognition, where we calculate the log likelihood rate (LLR) for the i-vector of each testing utterance and those of target L1s and select one with highest LLR as recognized L1.

4.4. Fusion of end-to-end and i-vector baseline system

We think that the end-to-end models and i-vector system might be able to compensate each other by learning different representations from the speech data. We have implemented score-level fusion by using the posterior probabilities generated at the output of the end-to-end neural networks and the log likelihood ratios computed using PLDA scoring model (normalized by z-score) as features to a Multilayer Perceptron (MLP) classifier to predict the L1 classes. The model contains two fully connected layers each with 200 hidden units and ReLU activation connected to a softmax layer with 11 hidden units. The network is optimized using SGD optimizer with an initial learning rate of 0.001, momentum of 0.9 and Nesterovs Accelerated Momentum update [33, 34]. The fusion model is trained using the development set mentioned in Section 4.1.

5. RESULTS AND DISCUSSION

We first compare the results obtained by different end-to-end systems using basic attention mechanism with the baseline i-vector system in Table 1. The performance of NLI systems are evaluated using accuracy and Unweighted Average Recall (UAR), same as the metrics used for ComParE challenge. All end-to-end models reported in Table 1 use basic attention mechanism as shown in Figure 3(a). All our end-to-end models perform significantly better than the majority-vote baseline which has an accuracy performance of 9% and 8.26% UAR. From our experimental results, we see the advantage of using a combination of CNN, GRU and DNN layers in the CGDNN architecture. The accuracy is improved significantly from 42.09% (RNN only) and 60.45% (CNN only) to 69.18% with the combined CGDNN system. This indicates that the CGDNN model is able to produce a feature representation that is more easily separable for L1 recognition.

The Listen, Attend, Identify (LAI) system outperforms all the end-to-end models achieving 70.45% accuracy and 70.87% UAR. The time resolution is reduced by $2^3 = 8$ times (for 3 pBLSTM layers) in the LAI model enabling us to focus on fewer number of time steps. The accuracy performance of the LAI model is 1.27% better than CGDNN model. However, from our experiments, time required for training the LAI model was significantly higher than the time required to train CGDNN model. Using three GPUs, training time for the LAI model was 6 times more than the training time required for the CGDNN model. The CGDNN model also benefited from using batch normalization layers allowing the model to converge in fewer number of epochs. Adding batch normalization layers to the LAI model had no effect on the convergence.

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**158**	2	3	2	1	3	3	2	6	0	6
CHI	0	**150**	0	0	0	0	1	2	0	0	0
FRE	20	14	**182**	29	2	18	13	12	8	2	9
GER	0	0	3	**162**	0	1	0	1	0	0	4
HIN	0	0	0	0	**129**	0	0	2	1	44	0
ITA	12	5	7	3	2	**165**	4	0	5	1	5
JPN	1	9	1	0	0	0	**157**	11	4	0	3
KOR	1	16	0	0	0	2	17	**165**	6	0	4
SPA	4	2	2	3	1	11	4	3	**169**	2	3
TEL	1	1	1	0	64	0	0	0	1	**151**	0
TUR	3	1	1	1	1	0	1	2	0	0	**166**

Fig. 4. Confusion matrix of the i-vector baseline system results on the test set of 11 L1s (rows: references; column: hypotheses)

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**134**	2	13	3	7	12	5	2	8	4	21
CHI	1	**148**	1	5	0	0	12	9	2	2	2
FRE	13	2	**143**	13	2	15	4	0	11	1	7
GER	1	3	11	**162**	1	9	2	1	7	0	3
HIN	3	1	1	2	**100**	1	1	0	4	31	1
ITA	4	4	15	6	2	**132**	1	3	7	1	5
JPN	6	8	3	0	2	0	**143**	14	9	1	3
KOR	5	24	1	5	2	4	18	**157**	11	1	3
SPA	14	5	4	2	3	19	10	9	**128**	4	3
TEL	4	0	1	1	81	0	2	1	4	**155**	4
TUR	15	3	7	1	0	8	2	4	9	0	**148**

Fig. 5. Confusion matrix of the LAI system (basic attention) results on the test set of 11 L1s (rows: references; column: hypotheses)

Confusion matrices for the results on the test set of 11 L1s using the i-vector baseline, the LAI and CGDNN systems are shown in Figures 4, 5 and 6 respectively. The most distinguishable L1s for the i-vector system are French (FRE), Spanish (SPA) and Turkish (TUR) which can all achieve an F1 score over 0.83. All three systems perform the worst on Hindi (HIN) recognition with F1 score less than 0.65. Hindi is most confused with the other Indian language in the corpus, Telugu (TEL). Although Telugu belongs to the Dravidian language family while Hindi belongs to the Indo-Aryan family, the two languages share many similarities in segmental pronunciation and prosody, which likely lead to similarities in the L2 English accents from speakers with these L1 backgrounds. However, both end-to-end systems outperform the baseline in Telugu recognition. While the i-vector baseline performs better for most L1s as compared to the LAI system, the performance of our best end-to-end system, i.e. LAI is close to the baseline performance for German (GER) and Chinese (CHI) recognition and outperforms the i-vector system on Telugu (TEL) recognition. Although the performance of

	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR
ARA	**137**	2	10	3	4	8	10	5	10	4	7
CHI	1	**161**	0	1	1	1	15	8	1	0	3
FRE	10	1	**139**	11	2	18	7	6	13	1	13
GER	4	6	18	**176**	2	7	2	10	10	1	9
HIN	4	1	0	0	**82**	1	0	0	0	32	0
ITA	16	0	15	2	4	**127**	3	1	21	3	6
JPN	2	8	2	0	1	4	**133**	12	5	0	4
KOR	1	12	1	2	2	1	12	**141**	8	1	2
SPA	8	3	9	3	1	22	7	10	**126**	4	9
TEL	2	1	0	0	99	1	1	0	2	**154**	1
TUR	15	5	6	2	2	10	10	7	4	0	**146**

Fig. 6. Confusion matrix of the LAI system (basic attention) results on the test set of 11 L1s (rows: references; column: hypotheses)

L1s	i-vector Baseline	LAI (basic attention)	CGDNN (basic attention)
ARA	0.79	0.67	0.685
CHI	0.75	0.74	0.805
FRE	0.91	0.715	0.695
GER	0.81	0.81	0.88
HIN	0.645	0.5	0.41
ITA	0.825	0.66	0.635
JPN	0.785	0.715	0.665
KOR	0.825	0.785	0.705
SPA	0.845	0.64	0.63
TEL	0.755	0.775	0.77
TUR	0.83	0.74	0.73

Table 2. The F1-score of the individual L1s on 11 L1s recognition

CGDNN system is lower than i-vector and LAI system, it is interesting to see that CGDNN outperforms both systems in German and Chinese recognition and performs on par with LAI on Telugu recognition outperforming the baseline.

Experimental results in Table 3 obtained using attention layer variants indicate that cross-layer attention performs slightly better relative to the basic attention model leading to approximately 1% relative improvement for both end-to-end models. Using divided-layer attention led to a degradation in end-to-end performance when compared with basic and cross-layer attention performance.

We report the number of parameters for the models in Table 4. We see that with relatively fewer parameters the end-to-end models are able to achieve reasonable performance when compared to the i-vector baseline. In particular, LAI system contains 7 times fewer parameters while CGDNN contains 43 times fewer parameters with respect to the i-vector system.

Each system shows superior performance in the recognition of specific L1s as seen from the individual confusion matrices in Figures 4, 5 and 6. Hence, we have tried fusion experiments where we use the prediction probabilities from the proposed end-to-end systems along with the LLRs from the i-vector system to train a new model to predict the L1 classes again. Table 5 illustrates the results obtained by fusion of baseline system with end-to-end models using basic and cross-layer attention mechanisms. All fusion systems reported in Table 5 can outperform the i-vector baseline system indicating that the end-to-end models may have learned complementary representations. The best fusion accuracy of 83.32% is obtained by fusion of LAI, CGDNN and i-vector systems where the end-to-end models use cross-layer attention mechanism. Fusion of the three systems where the end-to-end models use basic attention layer is just slightly worse in overall performance (83.18%). The basic attention layer is stronger at performing for complementary languages such as Chinese, German and Telugu recognition which the i-vector baseline is weak in recognizing, while for the models with cross-layer attention we saw additional improvements for Hindi, Italian, Japanese, Spanish and Turkish recognition in addition to the complementary language recognition. We see that for both attention variants in Table 5, CGDNN+baseline system performs better than LAI+baseline since as mentioned earlier in this section, CGDNN system performs well for a larger set of complementary languages (German, Chinese and Telugu) as compared to LAI system (Telugu).

Table 3. End-to-end systems accuracy with different attention layer.

Model	Basic	Cross-layer	Divided-layer
LAI	70.45	71.72	68.63
CGDNN	69.18	70.18	69.09

Table 4. Number of parameters for the models

Model	Parameters
i-vector baseline	$\sim 32M$
LAI	$\sim 4.46M$
CGDNN	$\sim 0.73M$

Table 5. The accuracy obtained by different fusion systems on the test set of 11 L1 corpus

Fusion system	Basic	Cross-layer
LAI + baseline	82.13	82.27
CGDNN + baseline	82.86	83.14
LAI + CGDNN + baseline	83.18	83.32

6. CONCLUSION

In this paper, we experimented with different end-to-end architectures and attention mechanism variants for automatic L1 recognition from raw features i.e. spectrogram of non-native speech. Our results indicate that our best attention-based neural network with much fewer parameters in the model can achieve a performance close to the conventional system. Additionally, in our model, feature representation learning and scoring can be done in a single system as opposed to the i-vector system. Furthermore, a fusion of the end-to-end system with the i-vector system can improve the baseline performance from 79.72% to 83.32% indicating that the three systems are able to capture complementary information from the data. We think that our results are promising to continue research in this direction for L1 recognition and we believe that the performance could be further improved with larger training data, an intrinsic property of deep learning to outperform the traditional i-vector system. Our future work will be on improving the end-to-end L1 recognition performance by investigating more advanced neural network architectures and attention variants and extending our system to handle L1 recognition of more languages.

7. REFERENCES

[1] C. Huang, E. Chang, J. Zhou, and K. F. Lee, "Accent modeling based on pronunciation dictionary adaptation for large vocabulary Mandarin speech recognition," in *Sixth International Conference on Spoken Language Processing*, 2000, pp. 803–806.

[2] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[3] K. Zechner, D. Higgins, R. Lawless, Y. Futagi, S. Ohls, and G. Ivanov, "Adapting the acoustic model of a speech recognizer for varied proficiency non-native spontaneous speech using read speech with language-specific pronunciation difficulty," in *In Tenth Annual Conference of the International Speech Communication Association*, 2009.

[4] B. W. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A.C. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native language," in *INTERSPEECH*, 2016, vol. 2016, pp. 2001–2005.

[5] G. Gosztolya, T. Grsz, R. Busa-Fekete, and L. Tth, "Determining native language and deception using phonetic features and classifier combination," in *INTERSPEECH*, 2016, pp. 2418–2422.

[6] Y. Qian, K. Evanini, X. Wang, D. Suendermann-Oeft, R. A. Pugh, P. L. Lange, H.R. Molloy, and F. K. Soong, "Improving sub-phone modeling for better native language identification with non-native english speech," in *INTERSPEECH*, 2017, pp. 2586–2590.

[7] Y. Qian, K. Evanini, P. L. Lange, R. A. Pugh, R. Ubale, and F. K. Soong, "Improving native language (L1) identifation with better VAD and TDNN trained separately on native and non-native english corpora," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 606–613.

[8] P. G. Shivakumar, S. N. Chakravarthula, and P. G. Georgiou, "Multimodal fusion of multirate acoustic, prosodic, and lexical speaker characteristics for native language identification," in *INTERSPEECH*, 2016, pp. 2408–2412.

[9] A. Abad, E. Ribeiro, F. Kepler, R. F. Astudillo, and I. Trancoso, "Exploiting phone log-likelihood ratio features for the detection of the native language of non-native english speakers," in *INTERSPEECH*, 2016, pp. 2413–2417.

[10] Y. Jiao, M. Tu, V. Berisha, and J. M. Liss, "Accent identification by combining deep neural networks and recurrent neural networks trained on long and short term features," in *INTERSPEECH*, 2016, pp. 2388–2392.

[11] A. Rajpal, T. B. Patel, H. B. Sailor, Patil H. A. Madhavi, M. C., and H. Fujisaki, "Native language identification using spectral and source-based features," in *INTERSPEECH*, 2016, pp. 2383–2387.

[12] G. Keren, J. Deng, J. Pohjalainen, and B. W. Schuller, "Convolutional neural networks with data augmentation for classifying speakers' native language," in *INTERSPEECH*, 2016, pp. 2393–2397.

[13] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[14] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[15] R. Prabhavalkar, K. Rao, T. N. Sainath, Johnson L. Li, B., and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[16] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, and N. Jaitly, "State-of-the-art speech recognition with sequence-to-sequence models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[17] R. Prabhavalkar, T. N. Sainath, Y. Wu, Chen Z. Nguyen, P., C. C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[18] T. N. Sainath, C. C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[19] C. C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[20] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Weiss R. J. Wu, Y., N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, and Q. Le, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017.

[21] J. Sotelo, S. Mehri, K. Kumar, Kastner Santos, J.F., A. K., Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *ICLR 2017 workshop*, 2017.

[22] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, and R. A. Saurous, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[23] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 569–5764.

[24] Y. Qian, R. Ubale, P. Lange, K. Evanini, and F. Soong, "From speech signals to semantics - tagging performance at acoustic, phonetic and word levels," in *Chinese Spoken Language Processing (ISCSLP), 2018 11th International Symposium (to appear)*. IEEE, 2018.

[25] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y Bengio, "Towards end-to-end spoken language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[26] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[27] S. X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification.," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv:1409.0473, 2014.

[29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *In International conference on machine learning*, 2015, pp. 2048–2057.

[30] F. A. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," arXiv preprint arXiv:1710.10470, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015, vol. 5.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*.

[33] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$," in *Doklady AN SSSR (translated as Soviet. Math. Docl.)*, 1983, vol. 269, pp. 543–547.

[34] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," arXiv preprint arXiv:1212.0901, 2012.