TOWARDS LANGUAGE-UNIVERSAL END-TO-END SPEECH RECOGNITION

Suyoun Kim∗
Dept. of Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA
suyoun@cmu.edu

Michael L. Seltzer
Speech & Dialog Research Group
Microsoft AI & Research
Redmond, WA 98052
mseltzer@microsoft.com

ABSTRACT

Building speech recognizers in multiple languages typically involves replicating a monolingual training recipe for each language, or utilizing a multi-task learning approach where models for different languages have separate output labels but share some internal parameters. In this work, we exploit recent progress in end-to-end speech recognition to create a single multilingual speech recognition system capable of recognizing any of the languages seen in training. To do so, we propose the use of a universal character set that is shared among all languages. We also create a language-specific gating mechanism within the network that can modulate the network’s internal representations in a language-specific way. We evaluate our proposed approach on the Microsoft Cortana task across three languages and show that our system outperforms both the individual monolingual systems and systems built with a multi-task learning approach. We also show that this model can be used to initialize a multilingual speech recognizer, and can be used to create a bilingual model for use in code-switching scenarios.

Index Terms— multilingual, language-universal

1. INTRODUCTION

As voice-driven interfaces to devices and information become mainstream, increasing the global reach of speech recognition systems becomes increasingly important. There are two primary challenges that arise in expanding the language coverage of a speech application. First, since conventional speech recognition systems require each model to be trained independently, as the number of supported languages grows, the effort required to train, deploy, and maintain so many models in a production environment will increase dramatically. In addition, for second- and third-tier languages with fewer resources available, issues with data scarcity arise. For each language, building a speech recognition system requires a large collection of transcribed speech recordings from many speakers to train an acoustic model, linguistic expertise to create a pronunciation dictionary, and vast amounts of text data to train a language model.

Over the years, prior work has attempted to address the issues faced by low-resource language via a transfer learning approach [1–8]. In these approaches, language-specific deep networks are trained in which the the parameters in the lower layers of the network are shared across languages. This approach can also be interpreted as an instance of multi-task learning, where information across tasks, i.e. languages, is shared to create a more informative internal representation, less prone to over-fitting. Another common approach for creating models in low resource languages is to adapt a neural acoustic model that has been well trained on a high-resource language. This is typically done by replacing output layer of the well-trained model and re-training the model to predict the targets of low-resource languages [2–5,9,10]. All of these models have been based on the conventional acoustic modeling strategy based on senones, and therefore still require a pronunciation lexicon to map words to phonemes and then senones.

Recently, new approaches to speech recognition that work in a so-called end-to-end manner have been proposed. In these systems, a neural network is trained to convert a sequence of acoustic feature vectors into a sequence of graphemes rather than senones. Unlike sequences of senone predictions, which need to be decoded using a pronunciation lexicon and a language model, the grapheme sequences can be directly converted to word sequences without any additional models or machinery. The end-to-end models proposed in the literature operate using a Connectionist Temporal Classification framework [11–15], an attention-based encoder-decoder framework [16–19], or both [20]. Thus far, the research in end-to-end systems has focused on monolingual scenarios and there have been few recent studies of grapheme-based models capable of recognizing multiple languages. One recent study proposed multilingual models that can jointly identify language [21], while another showed how a low-resource graphemic system can be initialized with a well-trained high-resource model, as was done for the senone-based models described previously [22].

In this paper, we make significant progress towards a language-universal speech recognizer by creating a single end-to-end system capable of recognizing any language it has been trained on. Our model exploits the recent progress in end-to-end approaches to output character sequences directly, without requiring pronunciation lexicons. As in other multilingual systems, we apply a transfer learning approach to share model parameters among multiple languages. The novel aspects of our proposed model are twofold. First, we use a single universal character set that can be shared among all languages rather than separate language-specific output layers. Second, we propose a language-specific gating mechanism in the network that can increase the network’s modeling power by using multiplicative interactions to modulate the network’s internal representations in a language-specific way. We evaluate our proposed model on the Microsoft Cortana personal assistant task and show that our system outperforms separate language-specific models as well as the conventional multi-task learning approach.

The remainder of this paper is organized as follows. In Section 2, we briefly review the end-to-end speech recognition paradigm. In Section 3, we describe approaches to multilingual CTC including multi-task learning and our proposed model that features end-to-end
learning with a universal character set and language-dependent gating units. In Section 4, we evaluate our proposed model through a series of experiments on the Microsoft Cortana task across three languages. Finally, we summarize our findings and discuss future avenues of research in Section 5.

2. END-TO-END MODELING USING CTC

In this work, we perform end-to-end speech recognition using a CTC-based approach with graphemes (characters) as the output symbols. With CTC, the neural network is trained according to a maximum-likelihood training criterion computed over all possible segmentations of the utterance’s sequence of feature vectors to its sequence of labels [11]. Core to CTC training is the presence of the blank symbol which can be interpreted as a “don’t care” symbol in the output label sequence. In CTC, the label sequences can contain blanks and repeated characters without penalty.

Given a sequence of acoustic feature vectors, \( x \) and the corresponding graphemic label sequence, \( y \), CTC trains the model to maximize the likelihood over all possible label sequences \( \Phi \).

\[
P(y|x) = \sum_{\pi \in \Phi(y)} P(\pi|x) \approx \sum_{\pi \in \Phi(y)} \prod_{t=1}^{T} P(\pi_t|x).
\]  

(1)

Likelihoods are computed using the well-known forward-backward algorithm and the gradient of the likelihood is used to update the network parameters using back propagation. After model training, decoding is performed in a greedy manner, where the most likely symbol at each time frame is hypothesized. The final sequence is obtained via a post-processing step where any character repetitions and blank symbols are removed from the output. Further improvements can be obtained by incorporating an external language model, at the character or word level [12–15].

In this work, we consider two neural architectures for a language-universal end-to-end model that both use the CTC objective function.

3. MULTILINGUAL END-TO-END MODELS

3.1. Multi-task CTC with language-specific character sets

Multi-task learning (MTL) has been proposed as a means for improving generalization performance in low-data scenarios. Training a model on multiple related tasks simultaneously serves as an inductive bias to improve the model’s performance. Many prior approaches to training acoustic models for low resource languages have used MTL as a means of sharing some of the acoustic model parameters across languages, e.g. all the parameters up to the final output layer \([1, 5, 6, 23]\). This forces the model to learn commonalities across languages which provides effective regularization and prevents over-fitting. Our MTL architecture is similar to previous senone-based models, where the lower layers are shared among the different languages while the output layer or layers are trained to be language specific. This architecture is shown in Figure 1a. Note that while the figure shows a language-specific output layer and final hidden layer, the choice of where to branch the model is a design choice that should be determined by experimentation.

Multi-task learning for CTC can be realized as a combination of the individual single-task objective functions. We define \( l \) to be an index over the different languages in the training data, \( x \) as a sequence of input feature vectors, and \( y_l \) the corresponding grapheme sequence. We can then define the CTC objective function for the \( l \)th language as

\[
F_l(\theta_s, \theta_l) \triangleq -\ln P(y_l|x, \theta_s, \theta_l)
\]  

(2)

where \( \theta_l \) are the language-specific model parameters, and \( \theta_s \) are the model parameters shared across languages. The multi-task objective function over all languages can then be defined as the summation of the individual negative log likelihoods of all \( L \) languages,

\[
\mathcal{F} \triangleq \sum_{l=1}^{L} F_l(\theta_s, \theta_l)
\]  

(3)

3.2. Multilingual CTC with a universal character set

In many instances, present-day languages evolved from a common ancestry. It is therefore natural that they share some common graphemes and phonemes. For example, the English character set is a subset of the Spanish character set and knowing one language helps to speak and write the other. With this as motivation, we propose a multilingual architecture that uses a “universal” output label set consisting of the union of all characters from the multiple languages, as illustrated in Figure 1b. Unlike the MTL approach, this is a single model with a universal character set that is trained on all multilingual data. Characters that are common across multiple languages are trained based on all relevant data while language-specific characters are only trained with data from that language.

Let \( Y_u \) be the universal label set. Given the training data from \( L \) multiple languages, including the acoustic input features \( x_u = \{x_1, \ldots, x_L\} \) and the corresponding graphemic label sequence \( y_u = \{y_1, \ldots, y_L\} \), where each label \( y_{u}^{(l)} \in Y_u \), the CTC loss for the models to be minimized is defined as,

\[
\mathcal{F}_U(\theta) \triangleq -\ln P(y_u|x_u, \theta)
\]  

(4)

In this model, we assume \( a \) priori that we know the language identity of the utterances in both training and decoding. We use this information to restrict the predictions to only those characters present in the corresponding language and mask the activations from the other irrelevant characters. Let the universal label set \( Y_u \) have \( K \) distinct labels. The label set for any particular language \( Y_l \) is a subset of \( Y_u (Y_l \subseteq Y_u) \). Given a language indicator, \( l \), we can mask out the activations from unwanted characters using a \( K \)-dimensional binary mask defined as

\[
M[l, k] = \begin{cases} 
0, & \text{if } k \notin Y_l \\
1, & \text{if } k \in Y_l 
\end{cases}
\]  

(5)
This mask is then applied to the network outputs to compute the log-likelihoods for the CTC objective function. Note that this masking operation is applied in both training and decoding.

3.3. Language-specific gating units

One possible drawback to the proposed universal character approach is that the same grapheme may have different underlying phonetic realizations in different languages. For example, the letter ñ in English typically corresponds to the sound /nh/ in English, but /iy/ in Italian. Thus, the model needs to adequately capture language-specific information in order to properly account for such differences. Adding a language identification feature as an auxiliary input to the model is the simplest way to do so. However, we found empirically that this only provides minimal improvement. Instead, we propose to use the language identity to modulate the network’s internal multilingual representations in a language-specific manner. To do so, the outputs of each hidden layer are processed by a series of language-dependent gates before being passed to the next layer in the model. Specifically, we first create a one-hot language indicator vector \( d_l \) for each language \( l \). Then, we compute the gate value based on the language indicator vector \( d_l \) and the current output values of \( h_i \), the \( i \)th hidden layer, as

\[
g(h_i, d_l) = \sigma(Uh_i + Vd_l + b)
\]

where \( U, V, \) and \( b \) are trainable parameters. The language-gated hidden activations are then calculated as

\[
\hat{h}_i = g(h_i, d_l) \odot h_i
\]

Finally, \( \hat{h}_i \) and \( d_l \) are concatenated and input to the next layer.

\[
\hat{h}_i = [\hat{h}_i : d_l]
\]

If the dimensions of \( h_i \) and \( d_l \) are \( n \) and \( m \), respectively, each language-specific gating layer requires \((n + m) \times n\) additional parameters.

4. EXPERIMENTS

4.1. Experimental corpora

We investigated the performance of the proposed language-universal model on English (EN), German (DE), and Spanish (ES) data from Cortana, Microsoft’s personal assistant. For each language, we used 150 hours of training data, 10 hours of validation data, and 10 hours of test data. We used 80-dimensional log-mel filterbank coefficients as acoustic features, derived from 25 ms frames with a 10 ms frame shift. We concatenate three consecutive feature vectors to input to the network and employ frame-skipping [24] which decimates the original frame rate by a factor of three. Thus, each feature vector is presented to the network exactly once. Following [15], we used a label symbol inventory consisting of the individual characters and their double-letter units. An initial capitalized letter rather than a space symbol was used to indicate word boundaries. This resulted in 81 distinct labels for English, 93 labels for German, and 97 labels for Spanish. Our universal label set for these three languages had 108 distinct labels and 81 overlapping labels. No pronunciation lexicon or language model was used in any of the experiments.

<p>| Table 1: Initial CER results for conventional single-task (stl), multi-task (mtl), and universal character set (univ) networks using monolingual and multilingual training data. |
|-----------------|--------------|---------|--------|--------|</p>
<table>
<thead>
<tr>
<th>Training Languages</th>
<th>Total Hrs</th>
<th>Model Arch</th>
<th>Test Lang</th>
<th>CER %</th>
<th>% Rel. Impr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>150</td>
<td>st</td>
<td>23.3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>DE + EN</td>
<td>300</td>
<td>stuniv</td>
<td>22.3</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>DE + EN + ES</td>
<td>450</td>
<td>univ</td>
<td>22.8</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>300</td>
<td>stuniv</td>
<td>15.8</td>
<td>32.2</td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>150</td>
<td>st</td>
<td>13.7</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ES + EN</td>
<td>300</td>
<td>stuniv</td>
<td>13.1</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td>ES + EN + ES</td>
<td>450</td>
<td>univ</td>
<td>12.9</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>300</td>
<td>stuniv</td>
<td>11.7</td>
<td>14.4</td>
<td></td>
</tr>
</tbody>
</table>

4.2. Training and decoding

Our language-universal encoder was a 4-layer Bidirectional Long Short-Term Memory (BLSTM) network [25, 26] with 320 cells in each layer and direction. A 320-dimensional linear projection layer followed each BLSTM layer. All the weights in the models were initialized with a uniform distribution in the range of [-0.05, 0.05], and were trained using stochastic gradient descent with momentum. We used a learning rate of 0.0004 and gradient-clipping threshold per sample of 0.0003. Early stopping on the validation set was used to select the best model. For the decoding, the most likely sequence of characters was generated by the model in a greedy manner. The final output sequence was then obtained by removing any blank symbols or repetitions of characters from the output and replacing any capital letter with a space and its lowercase counterpart.

4.3. Gated language-universal end-to-end models

We first evaluated both multi-task and language-universal architectures on German and Spanish data, along with conventional monolingual training. As seen in Table 1, we obtained a small performance gain over the monolingual model by using multilingual training data. In addition, we observed that models trained with multi-task learning with language-specific output labels and models trained with a universal label set performed comparably, demonstrating the potential of a universal label set. However, with a conventional BLSTM topology, the benefit gained from the multilingual data was far less than simply doubling the amount of training data from the target language to 300 hours. More concerning, the performance actually degraded when we increased the multilingual data from two to three languages. We also extensively evaluated various configurations for multi-task learning, where all or only some of the layers were shared, and found no significant improvements over the results reported in Table 1. These initial results indicate that simply training conventional models with multilingual data is not very helpful. However, we note that these results may differ in extremely low resource scenarios where the data available per language could be an order of magnitude smaller.

We next evaluated the proposed language-universal model with language-specific gating. The results in Table 2 show that the proposed gated model significantly outperforms the monolingual model as well as the conventional multilingual models. Our model showed 10.7%, 11.4%, and 14.1% relative improvement in CER compared to a monolingual for EN, DE, and ES respectively. The relative im-
Improvements in WER, which was calculated without using any language model, were 7.0%, 8.6%, and 11.1%, for EN, DE, and ES respectively. Furthermore, when language-specific gating is used, additional improvement is obtained when the number of training languages increased from two to three.

Figure 2 shows the relative improvement over the baseline WER obtained by adding language information to the model in various ways. The leftmost bar shows the result of simply augmenting the input to each layer with the one-hot language vector \( d_l \). The next three bars show the performance of different gating functions driven by the current hidden state, the language identity, or both, respectively. Finally, the rightmost bar shows the approach shown in Equations 7-8. As the figure indicates, all approaches provide gains over the baseline model, but the proposed approach results in the largest improvement. We also investigated the effect of the gating mechanism on different layers and observed that the best performance was obtained when gating is applied to every layer (not shown). Note that we used 7.3M parameters for our proposed model and 6.8M parameters for the baseline.

![Graph of relative improvement over baseline](image)

**Fig. 2:** The relative improvement in WER obtained by using language identification information as an auxiliary input and/or a gating mechanism.

### 4.4. Initial model for subsequent language-specific models

While the primary goal of this work was to create a language-universal model, we found that this model is also a good initial model for creating a language-specific monolingual model when the training data is limited. Table 3 shows the CER on the DE test set obtained from different pre-training strategies used to initialize the model prior to fine-tuning on the DE training data. Notably, initializing with the gated language-universal model outperformed all of the other approaches, including bootstrapping from an EN model trained with significantly more data. The second best performance was obtained by the gated language-universal model directly, even without further fine-tuning on the DE training data.

![Table 3: CER obtained for a monolingual model with different pre-training strategies](image)

### 4.5. Bilingual end-to-end models

One promising aspect of a universal output character set is the potential to create end-to-end systems that can dynamically code-switch between languages. This can be done by training a model on bilingual training data with the union of the output symbols of the two languages while omitting the language-specific aspects of the model. Following this approach, we created a bilingual model from 150 hours of English (EN) and 150 hours of Spanish (ES) using the union of the character sets from the two languages.

Because a test set of bilingual mixed-code speech was not available, we evaluated this model on the EN and ES test sets separately. Table 4 shows that bilingual models perform as well as the monolingual models even if no language identification information is provided to the network. Thus, this approach is very promising as a method for creating end-to-end systems capable of code-switching during decoding.

![Table 4: WER on ES and EN test sets obtained from monolingual and bilingual end-to-end systems](image)

## 5. CONCLUSION

We proposed a language-universal end-to-end speech recognition system capable of recognizing speech from any language seen in training. Key aspects of this model include the use of a universal character set and language-specific gating units. Because this model can support multiple languages, it has the potential to simplify model deployment in production environments. In addition, no pronunciation lexicon is required, which is beneficial for second- and third-tier languages, where such resources may be unavailable or limited. Our model was shown to outperform models trained on monolingual training data, as well as with the multi-task learning approach employed in previous work. Moreover, we found that our language-universal speech recognizer is a strong initial model for subsequent monolingual training, and demonstrated the potential of this system for decoding speech in the presence of code-switching. Moving forward, we plan to explore how to add a new language to an already-trained multilingual system and evaluate the performance improvements that can be obtained by using an external language model.
6. REFERENCES


