AN END-TO-END LANGUAGE-TRACKING SPEECH RECOGNIZER FOR MIXED-LANGUAGE SPEECH

Hiroshi Seki[*]*, Shinji Watanabe, Takaaki Hori, Jonathan Le Roux, John R. Hershey*

Mitsubishi Electric Research Laboratories (MERL), Cambridge MA, USA

ABSTRACT

End-to-end automatic speech recognition (ASR) can significantly reduce the burden of developing ASR systems for new languages, by eliminating the need for linguistic information such as pronunciation dictionaries. This also creates an opportunity to build a monolithic multilingual ASR system with a language-independent neural network architecture. In our previous work, we proposed a monolithic neural network architecture that can recognize multiple languages, and showed its effectiveness compared with conventional language-dependent models. However, the model is not guaranteed to properly handle switches in language within an utterance, thus lacking the flexibility to recognize mixed-language speech such as code-switching. In this paper, we extend our model to enable dynamic tracking of the language within an utterance, and propose a training procedure that takes advantage of a newly created mixed-language speech corpus. Experimental results show that the extended model outperforms both language-dependent models and our previous model without suffering from performance degradation that could be associated with language switching.

***Index Terms*—** End-to-end ASR, multilingual ASR, language-independent architecture, language identification, hybrid attention/CTC

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) has recently proven its effectiveness by reaching levels of accuracy equivalent to the state-of-the-art conventional hybrid systems [1, 2, 3, 4], while surpassing them in terms of ease of development. Conventional ASR systems require language-dependent resources such as pronunciation dictionaries and word segmentation, which are incorporated into models with phonemes as an intermediate representation. These resources are developed by hand and so they carry two disadvantages: first, they may be error-prone or otherwise sub-optimal, and second, they greatly increase the effort required to develop ASR systems, especially for new languages. The use of language-dependent resources thus particularly complicates the development of multilingual recognition systems. End-to-end ASR systems, in contrast, directly convert input speech feature sequences to output label sequences (mainly sequences of characters or tokens composed of n-gram characters in this paper) without any explicit intermediate representation of phonetic/linguistic constructs such as phonemes or words. Their main advantage is that they avoid the need for hand-made language-dependent resources.

There have been several prior studies on multilingual/language-independent ASR [5, 6, 7]. In the context of a deep neural network (DNN)-based multilingual system, the DNN is used to compute language independent bottleneck features. In such models it is necessary to prepare language-dependent back end components like pronunciation dictionaries and language models. In addition, the uttered language has to be predicted in order to cascade language-independent and language-dependent modules [8, 9].

In our previous work [10], we proposed a monolithic ASR system, with a fully language-independent neural network architecture, that can recognize speech and identify language jointly in ten different languages: English, Japanese, Mandarin, German, Spanish, French, Italian, Dutch, Portuguese, and Russian. Unlike a conventional language-dependent system, all parameters can be shared across languages, including those in the output layer, because the output set is the union of the grapheme sets of the multiple languages. This monolithic end-to-end ASR system has three advantages: first, the monolithic architecture obviates the need for language-dependent ASR modules and a separate language identification module; second, the end-to-end architecture makes it unnecessary to prepare a hand-crafted pronunciation dictionary; and third, the shared network enables the learning of better feature representations even for low-resource languages. However, in the previous setup, we trained the model using data consisting of one language per utterance. This setup does not guarantee the proper handling of a language switch within an utterance because it does not need to handle such a phenomenon at training time. In linguistics, the alternating of languages by a speaker is called code-switching. Speech recognition is thought to be more difficult with code-switching, especially if it is intrasentential (within sentences).

In this paper, we extend our language-independent architecture to enable the flexibility to switch languages within an utterance while maintaining the original feature of joint language identification and speech recognition. There exist a few corpora with code-switching [11, 12], but their size and the number of language combinations are limited. Therefore, we generate a new corpus in order to evaluate our proposed model against changes of language within an utterance. We further propose a method inspired by curriculum learning to efficiently train our system. This extension is advantageous in the recognition of utterance with intrasentential code-switching and further eliminates the need for an accurate voice activity detection (VAD) system for the segmentation of languages.

2. LANGUAGE-INDEPENDENT ARCHITECTURE WITH FLEXIBLE CODE-SWITCHING

2.1. Augmented character set

A key idea in our original language-independent end-to-end system [10] is to consider as the set of output symbols an augmented character set including the union of character sets appearing in all the target languages, i.e., $\mathcal{U} = \mathcal{U}^{\text{EN}} \cup \mathcal{U}^{\text{JP}} \cup \cdots$, where $\mathcal{U}^{\text{EN/JP/}\cdots}$ is a character set of a specific language. By using this augmented character set, likelihoods of character sequences can be computed for

[*]This work was done while H. Seki, Ph.D. candidate at Toyohashi University of Technology, Japan, was an intern at MERL.

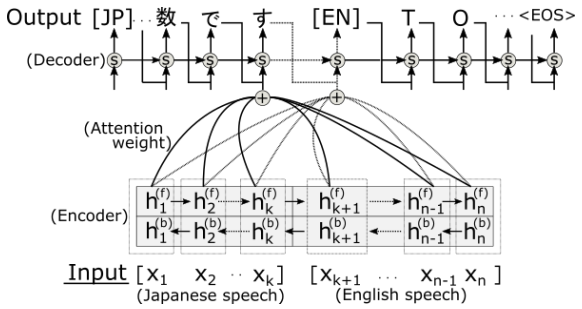

Fig. 1. The proposed language-independent architecture. The system repeats the prediction of language ID and speech recognition given consecutive multilingual speech.

any language, without requiring a separate language identification module. The network is trained to automatically predict the correct character sequence for the target language of each utterance. The use of the union, as opposed to using a unique character set for each language, eliminates the duplication of output symbols that occur in multiple languages, and yields a more compact model representation with reduced computational cost.

Table 1 summarizes the number of distinct characters/tokens including special symbols (e.g., <blank> used in CTC, end of sentence <eos>, space <space>). For English, in order to handle the relatively long sentences in the WSJ corpus, we expand the alphabet character set to 201 by adding tokens corresponding to up to 5-gram character sequences frequently appearing in the WSJ text corpus. This makes the output length L shorter, in order to reduce computational cost and GPU memory usage.

Table 1. Numbers of distinct characters/tokens (or character/token vocabulary size) for each language and the union of all languages.

EN	JP	CH	DE	ES	FR	IT	NL	PT	RU	Union
201	3,315	3,653	54	33	59	37	32	41	35	5,510

2.2. Joint language identification and speech recognition

Instead of letting the system implicitly predict the target language of the utterance, we make the prediction of the language ID an explicit part of the system by further augmenting the set of output tokens to include the language ID, i.e., $\mathcal{U}^{\text{final}} = \mathcal{U} \cup \{\text{EN}, \text{JP}, \cdots\}$. In our previous paper, the network first predicts a language ID, $k \in \{\text{EN}, \text{JP}, \cdots\}$, only once. Instead of a posterior distribution $p(C|X)$ where $C = (c_1, \ldots, c_L)$ is a sequence of characters in $\mathcal{U}$ and X is a sequence of acoustic features, the system models the joint distribution $p(k, C|X)$ of the language ID and character sequence as that of an augmented sequence $C' = (k, C)$ where $c'_0 = k$ and $c'_l = c_l, \forall l > 0$. This is formulated using the probabilistic chain rule as follows:

$$p(k, C|X) = p(k) \prod_l p(c_l|k, c_1, \ldots, c_{l-1}, X).$$

In this paper, we extend the architecture by removing the restriction to output a unique language ID k once and for all at the beginning of the sequence, thus allowing the network to output multiple language IDs throughout. For a sequence $C = (c_l)$ of characters in $\mathcal{U}^{\text{final}}$, we denote by $l_1, \ldots, l_N$ the indices of the characters $k_n = c_{l_n}$ in C that are language IDs (i.e., $k_n \in \{\text{EN}, \text{JP}, \cdots\}$). The system now models the joint distribution of language IDs and characters as

$$p(C|X) = \prod_n p(k_n|c_1, \ldots, c_{l_n-1}, X) \prod_{l=l_n+1}^{l_{n+1}-1} p(c_l|k_n, c_1, \ldots, c_{l-1}, X).$$

Table 2. Details of original language-dependent corpora without code-switching and generated corpus with code-switching. The coverage of each original corpus in the generated corpus is shown after the slash.

Corpus	Tasks	Length (h) / Coverage(%)	
		Original	Generated
WSJ English (**EN**)	Training	81.5 / —	87.4 / 63.6
	Development	1.1 / —	0.8 / 54.4
	Evaluation	0.7 / —	0.5 / 57.6
CSJ Japanese (**JP**)	Training	216.3 / —	149.1 / 49.1
	Development	6.6 / —	5.2 / 59.2
	Evaluation	5.2 / —	4.8 / 66.9
HKUST Mandarin (**CH**)	Training	170.1 / —	114.9 / 74.1
	Development	4.8 / —	6.2 / 66.0
	Evaluation	4.9 / —	6.9 / 64.3
Voxforge German (**DE**)	Training	45.7 / —	64.6 / 54.2
	Development	5.5 / —	6.9 / 68.0
	Evaluation	5.6 / —	6.8 / 68.4
Voxforge Spanish (**ES**)	Training	40.3 / —	61.6 / 78.6
	Development	3.2 / —	2.9 / 70.3
	Evaluation	7.0 / —	4.0 / 47.5
Voxforge French (**FR**)	Training	29.6 / —	57.9 / 88.4
	Development	4.0 / —	4.4 / 78.5
	Evaluation	3.6 / —	3.9 / 78.1
Voxforge Italian (**IT**)	Training	15.8 / —	35.7 / 92.9
	Development	2.1 / —	2.0 / 72.8
	Evaluation	2.0 / —	1.9 / 72.6
Voxforge Dutch (**NL**)	Training	8.4 / —	23.6 / 96.7
	Development	1.1 / —	1.3 / 82.3
	Evaluation	1.1 / —	1.3 / 81.5
Voxforge Portuguese (**PT**)	Training	3.0 / —	9.0 / 98.3
	Development	0.4 / —	0.5 / 80.2
	Evaluation	0.3 / —	0.4 / 90.6
Voxforge Russian (**RU**)	Training	12.0 / —	18.5 / 74.3
	Development	1.7 / —	1.0 / 50.2
	Evaluation	1.3 / —	1.0 / 63.9

Table 3. Number of utterances with $n_{\text{concat}} = 2$ in the evaluation set according to the number of included speakers and languages.

		# languages	
		1	2
# speakers	1	85	28
	2	868	2707

When recognizing an utterance with code-switching, the network can switch the language of the output sequence. Figure 1 shows an example of forward computation. The bi-directional encoder network computes hidden representations by taking as input acoustic features consisting of Japanese and English speech. The decoder network predicts language ID "[JP]" followed by a Japanese character sequence. After decoding the first Japanese character sequence, the network predicts the language ID that matches the character sequence that follows, here "[EN]". There is no boundary or indicator in the hidden representations between the languages. This forces the network to predict the language ID for variable length segments.

2.3. Data generation for multilingual speech

In this section, we describe the generation of a new corpus with code-switching by integrating existing language-dependent corpora without code-switching. We selected utterances from language-dependent corpora while paying attention to the coverage of selected utterances and the variation of language transitions as described further below. The selected utterances are concatenated to form a single utterance in the generated corpus (we concatenated whole utterances for simplicity, but using parts of utterances may be more

Table 4. Character Error Rates (CERs, %) on the original evaluation set without code-switching.

	Training with code-switching	HKUST	WSJ	CSJ	Voxforge							Avg.
		CH	EN	JP	DE	ES	FR	IT	NL	RU	PT	
Language-dependent	×	35.1	7.4	13.2	5.2	50.8	26.5	14.3	25.5	49.4	52.2	28.0
Language-independent	×	33.3	5.1	10.9	5.5	31.9	19.8	11.1	18.6	33.1	28.0	19.7
	✓ (flat start)	41.2	6.6	14.8	6.2	33.4	21.3	10.6	18.2	37.6	28.0	21.8
	✓ (retrain)	31.8	5.0	10.9	5.3	33.1	19.2	9.8	17.3	34.7	26.9	19.4

realistic in code-switching applications). This procedure is repeated until the duration of the generated corpus reaches that of the union of the original corpora. Table 2 shows the details of the original language dependent corpora (without code-switching) and the generated corpus (with code-switching). The original corpora are based on WSJ [13, 14], CSJ [15], HKUST [16], and Voxforge (German, Spanish, French, Italian, Dutch, Portuguese, Russian) [11], which are the same as in our previous paper, except for CSJ and HKUST: speech of HKUST was up-sampled from 8 kHz to 16 kHz, but speed perturbation was not applied; only academic lectures were used in CSJ. The column "original" shows the corpora before concatenation, with their duration indicated before the slash. The column "generated" shows the duration and coverage of the generated corpus. The coverage shows the ratio of the duration selected for concatenation to the utterances in the original corpus. We can see that the duration of large-scale corpora (e.g., CSJ and HKUST) is decreased while that of smaller corpora (e.g., Voxforge) is increased.

Algorithm 1 Generation of code-switching corpus

$N_{\text{concat}} \Leftarrow$ maximum number of utterances to concatenate.
$N \Leftarrow$ number of languages.
$D \Leftarrow$ duration of the union of the original corpora.
$n_{\text{reuse}} \Leftarrow$ maximum number of times same utterance can be used.
for $i \leftarrow 1$ to N **do**
  $P(\text{lang}_i) = \frac{1}{2} \frac{\text{duration of lang}_i}{\sum_j \text{duration of lang}_j} + \frac{1}{2N}$
  $P(\text{utter}_{\text{lang}_i,k}) = \frac{1}{\text{number of utterances in lang}_i}$
end for
while duration(*generated corpus*) $\leq D$ **do**
  for $n_{\text{concat}} \leftarrow 1$ to N_{concat} **do**
    for $i \leftarrow 1$ to n_{concat} **do**
      Sample language lang_i and utterance $\text{utter}_{\text{lang}_i,k}$, resampling if $\text{utter}_{\text{lang}_i,k}$ already selected n_{reuse} times.
    end for
    Concatenate n_{concat} utterances.
    Add to *generated corpus*.
  end for
end while

Algorithm 1 shows the details of the generation procedure. We first define probabilities to sample languages and utterances. The probability of sampling a language is proportional to the duration of its original corpus, with a constant term $1/N$ added to alleviate the selection bias caused by data size. We set a maximum number N_{concat} of utterances to concatenate, 3 in our experiment. For each number n_{concat} between 1 and N_{concat}, we create a concatenated utterance consisting of n_{concat} utterances from the original corpora, by sampling n_{concat} languages and utterances based on their sampling probabilities. In order to maximize the coverage of the original corpora, we prevent utterances from being reused too much by introducing a maximum usage count, n_{reuse}, set to 5 for the training set, and 2 for the development and evaluation sets. We use this procedure to generate a training set, a development set, and an evaluation set.

There is a concern in the generated corpus that the change of speaker fully synchronizes with the change of language. Table 3 shows the number of utterances in the evaluation set for which $n_{\text{concat}} = 2$ (concatenations of two utterances), according to the number of included speakers and languages. 85 samples concatenate utterances from the same language by the same speaker. 868 samples concatenate utterances from the same language by different speakers. 28 samples concatenate utterances from different languages by the same speaker (this can occur in the Voxforge corpus). In our experiments, we separately evaluate these 28 utterances, uttered by 10 speakers, as a more realistic scenario, e.g., recognition of foreign named entities, which is referred to as *real*. The number of such utterances in the training set is 2.

2.4. Training Procedure

In our experiments, we consider two training procedures. In the *flat start* procedure, the model is trained only using the generated corpus, from scratch. In the *retrain* procedure, the model is trained in two steps, using both the original and generated corpora as follows. We first train the model using the training data without code-switching (i.e., the original corpora), then continue the training using the data with code-switching (generated corpus). We consider these two steps for the following reasons. First, the model trained by the data without code-switching is a good starting point for the training of the arguably more difficult data with code-switching, in the spirit of curriculum learning [17]. Second, we allowed the data generation algorithm to select duplicated utterances in order to increase the ratio of imbalanced low-resource languages in the dataset. However, this property causes a decrease in coverage. The two step training alleviates this problem.

3. EXPERIMENTS

3.1. Setup

We built language-dependent and language-independent end-to-end systems with the same hybrid attention/connectionist temporal classification (CTC) network architecture [3] and hyperparameters as in our previous paper [10]. The language-dependent model uses a 4-layer encoder network, while the language-independent model has a deeper 7-layer encoder network.

We used 80-dimensional mel filterbank features concatenated with 3-dimensional pitch features as implemented in Kaldi [18]. For the language-independent models, the final softmax layers in both CTC and attention-based branches had 5,520 dimensions (i.e., $|\mathcal{U}^{\text{final}}| = 5,520$). For each language, we trained a language-dependent ASR model, where the dimension of the final softmax layers was set to the number of distinct characters/tokens for that language as shown in Table 1. This paper strictly followed an end-to-end ASR concept, and did not use any pronunciation lexicon, word-based language model, GMM/HMM, or DNN/HMM. Our hybrid attention/CTC architecture was implemented with Chainer [19].

3.2. Results

Table 4 shows the character error rates (CERs) of the trained language-dependent and language-independent end-to-end ASR

Table 5. CERs (%) on generated evaluation set.

Training with code-switching	# concatenated utter. 1	2	3	Avg.	*real*
×	21.1	31.5	38.6	32.2	30.5
✓ (flat start)	23.0	21.3	20.8	21.5	27.8
✓ (retrain)	21.2	19.3	18.6	19.4	26.4

Table 6. CERs (%) with oracle code-switching on generated evaluation set.

Training with code-switching	# concatenated utter. 1	2	3	Avg.	*real*
×	21.1	19.3	18.6	19.4	23.7
✓ (flat start)	23.0	21.1	20.6	21.3	26.7
✓ (retrain)	21.2	19.1	18.3	19.2	26.4

systems on the original evaluation set, where each sample consists of a single utterance in a single language. The Avg column shows the macro average CER for the 10 languages. As in our previous work, the language-independent models showed better performance on average than the language-dependent models. Further analyzing the bottom three rows, we can see the influence on CERs of the choice of training corpora and training procedure. The flat start training of the model on the generated corpus leads to slightly worse performance than the model trained on the original corpora, while still outperforming the language-dependent models. The retrained model outperforms both the flat start model and the model trained on the original corpora, even though evaluation is performed on speech with no code-switching.

Table 5 shows the CERs on the generated evaluation set, which includes code-switching. The table reports the CERs for each number of concatenated utterances and for the realistic scenario, in addition to the average CERs. The model trained on the original corpora obtains an average CER of 32.2%. The use of the generated code-switching corpus and the two-step training procedure improved the average performance by 12.8% absolute, mainly improving the performance for concatenated utterances. The model is able to properly switch between languages during the decoding process, as can be seen from the consistent performance for different numbers of concatenated utterances. The *real* column is a set of utterances where a single speaker speaks two languages. The CERs also decreased by training the model using code-switching corpus.

For further analysis, we considered the performance that would have been obtained by each of our systems assuming oracle code-switching: for a sample X generated by concatenating utterances $X_1, \ldots, X_{n_{\text{concat}}}$ with reference transcripts $C_1, \ldots, C_{n_{\text{concat}}}$, we generate a pseudo-transcription $(\tilde{C}_1, \ldots, \tilde{C}_{n_{\text{concat}}})$ by concatenating the estimated transcriptions $\tilde{C}_n$ obtained by the system on X_n alone. Table 6 shows the results. By comparing them with those in Table 5, we see that the performance of the *retrain* model is roughly the same whether it is given oracle knowledge of the code-switching points or not, showing its robustness in handling utterances with changes in language. On the contrary, the CER of the model without code-switching increases significantly when the model is not given oracle knowledge of the switching points.

Table 7 shows language identification error rates (LERs). The LER was calculated by computing the edit distance between the predicted language IDs and corresponding reference IDs, discarding non-ID characters. The results in parenthesis with "or." are oracle results obtained as above. Similarly to CERs, the LER of the model trained on the corpus without code-switching worsened dramatically to 51.8%, mainly due to deletion errors: the model output

Table 7. LERs (%) on original and generated evaluation sets.

Training with code-switching	Evaluation set (code-switching: ×/✓) original (×)	generated (✓)	*real* (✓)
×	1.9	51.8 (or.: 1.5)	51.8 (or.: 5.4)
✓ (flat start)	2.1	6.6 (or.: 1.7)	14.3 (or.: 5.4)
✓ (retrain)	2.0	8.5 (or.: 1.6)	10.7 (or.: 7.1)

Table 8. Examples of recognition results.

Model w/o code-switching
[IT] P O I N T O N A N D E ␣ S C H E D O ␣ A ␣ F A N O T O ␣ C H E ␣ M A ␣ C H E G I R A V A ␣ I N A C A T T A N D I R ␣ C H E D O ␣ W E ' R E ␣ S E E I N G ␣ A ␣ R E S P O N S E ␣ T H E I R ␣ I G H T ␣ D I R E C T I O N ␣ B U T ␣ I T ' S ␣ C L E A R L Y ␣ T E M P O R A R E D ␣ B Y ␣ N A T U R A L ␣ C O N S E R V A T I V E S ␣ I N ␣ W I G H T ␣ N O H W H D O O N Z E I D A T O N I E R ␣ D A N ␣ V O L G E N ␣ Z O C I S

Model w/ code-switching
[JP] ボ イ ン ト な ん で す け と そ の 時 は ま ー こ ち ら は い な か っ た ん で す け と **[EN]** ␣ W E ' R E ␣ S E E I N G ␣ A ␣ R E S P O N S E ␣ I N ␣ T H E ␣ R I G H T ␣ D I R E C T I O N ␣ B U T ␣ I T ' S ␣ C L E A R L Y ␣ T E M P O R A R E D ␣ B Y ␣ N A T U R A L ␣ C O N S E R V A T I V E S ␣ I N ␣ R I G H T ␣ N O W **[NL]** D O O N ␣ Z I J ␣ D A T ␣ N I E T ␣ D A N ␣ V O L G E N D E ␣ Z A M P Z I S

Reference
[JP] ボ イ ン ト な ん で す け と そ の 時 は ま ー ク ジ ラ は い な か っ た ん で す け と **[EN]** W E ' R E ␣ S E E I N G ␣ A ␣ R E S P O N S E ␣ I N ␣ T H E ␣ R I G H T ␣ D I R E C T I O N ␣ B U T ␣ I T ' S ␣ C L E A R L Y ␣ T E M P E R E D ␣ B Y ␣ N A T U R A L ␣ C O N S E R V A T I S M ␣ R I G H T ␣ N O W **[NL]** D O E N ␣ Z I J ␣ D A T ␣ N I E T ␣ D A N ␣ V O L G E N ␣ S A N C T I E S

a single language ID in almost all multilingual utterances. However, training on the code-switching corpus significantly recovers the LER by forcing the network to predict language IDs based on variable length segments. We think this ability to predict language IDs with high accuracy within an utterance is advantageous in later text-based post-processing.

We briefly analyse the results for the *real* scenario. The models with code-switching again showed similar CERs in the oracle and non-oracle results: while we should be cautious to draw conclusions given the small size (28 utterances) of the *real* set, this seems to show that the models are not relying on the change of speaker as a hint to switch languages. Indeed, the LER for *real* is only slightly worse than for the whole generated set in the non-oracle case, while it is much worse in the oracle case, hinting at *real* being a more difficult than an average set of utterances for language ID. We shall note that the higher CERs for *real* can be partly explained by the particular mix of languages in that subset: if we adjust the language distribution of the generated set (excluding *real*) to match that of *real*, the *retrain* model leads to a CER of 21.2% in both oracle and non-oracle conditions.

Table 8 shows transcription examples generated by our models. The utterance consists of Japanese, English, and Dutch. The model without code-switching cannot predict neither the correct language IDs nor the use of the Japanese character set. We can observe that the model with code-switching recognized multilingual speech with low CER.

4. SUMMARY

We extended our monolithic multilingual ASR system to enable the flexibility of switching between languages within an utterance. Since the resources of code-switching corpora are limited, we artificially generated a large-scale corpus in which multiple languages appear within the same utterance. Experimental results showed that our proposed model obtains the best performance in the recognition of speech which includes code-switching. In addition, comparison with oracle results shows that the performance is roughly the same whether an utterance features a single language or multiple languages. Future work will consider evaluation on real-world code-switching data.

5. REFERENCES

[1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[2] Hagen Soltau, Hank Liao, and Hasim Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[3] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Interspeech*, 2017, pp. 949–953.

[4] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[5] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[6] Zoltan Tuske, David Nolden, Ralf Schluter, and Hermann Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7854–7858.

[7] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, 2012, pp. 336–341.

[8] Javier Gonzalez-Dominguez, David Eustis, Ignacio Lopez-Moreno, Andrew Senior, Françoise Beaufays, and Pedro J Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 749–759, 2015.

[9] Shigeki Matsuda, Xinhui Hu, Yoshinori Shiga, Hideki Kashioka, Chiori Hori, Keiji Yasuda, Hideo Okuma, Masao Uchiyama, Eiichiro Sumita, Hisashi Kawai, et al., "Multilingual speech-to-speech translation system: Voicetra," in *IEEE International Conference on Mobile Data Management (MDM)*, 2013, vol. 2, pp. 229–233.

[10] Shinji Watanabe, Takaaki Hori, and John R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[11] "VoxForge," http://www.voxforge.org/.

[12] Emre Yilmaz, Maaike Andringa, Sigrid Kingma, Jelske Dijkstra, Frits Van der Kuip, Hans Van de Velde, Frederik Kampstra, Jouke Algra, Henk Van den Heuvel, and David Van Leeuwen, "A longitudinal bilingual Frisian-Dutch radio broadcast database designed for code-switching research," in *Language Resources and Evaluation Conference*, 2016, pp. 4666–4669.

[13] Linguistic Data Consortium, "CSR-II (wsj1) complete," *Linguistic Data Consortium, Philadelphia*, vol. LDC94S13A, 1994.

[14] John Garofalo, David Graff, Doug Paul, and David Pallett, "CSR-I (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, vol. LDC93S6A, 2007.

[15] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous speech corpus of Japanese," in *International Conference on Language Resources and Evaluation (LREC)*, 2000, vol. 2, pp. 947–952.

[16] Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff, "HKUST/MTS: A very large scale Mandarin telephone speech corpus," in *Chinese Spoken Language Processing*, pp. 724–735. Springer, 2006.

[17] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[18] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 2494–2498.

[19] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS*, 2015.