Qin *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2022) 2022:2
https://doi.org/10.1186/s13636-021-00233-4

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Improving low-resource Tibetan end-to-end ASR by multilingual and multilevel unit modeling

Siqing Qin[1], Longbiao Wang[1*], Sheng Li[2*], Jianwu Dang[1,3] and Lixin Pan[4]

Abstract

Conventional automatic speech recognition (ASR) and emerging end-to-end (E2E) speech recognition have achieved promising results after being provided with sufficient resources. However, for low-resource language, the current ASR is still challenging. The Lhasa dialect is the most widespread Tibetan dialect and has a wealth of speakers and transcriptions. Hence, it is meaningful to apply the ASR technique to the Lhasa dialect for historical heritage protection and cultural exchange. Previous work on Tibetan speech recognition focused on selecting phone-level acoustic modeling units and incorporating tonal information but underestimated the influence of limited data. The purpose of this paper is to improve the speech recognition performance of the low-resource Lhasa dialect by adopting multilingual speech recognition technology on the E2E structure based on the transfer learning framework. Using transfer learning, we first establish a monolingual E2E ASR system for the Lhasa dialect with different source languages to initialize the ASR model to compare the positive effects of source languages on the Tibetan ASR model. We further propose a multilingual E2E ASR system by utilizing initialization strategies with different source languages and multilevel units, which is proposed for the first time. Our experiments show that the performance of the proposed method-based ASR system exceeds that of the E2E baseline ASR system. Our proposed method effectively models the low-resource Lhasa dialect and achieves a relative 14.2% performance improvement in character error rate (CER) compared to DNN-HMM systems. Moreover, from the best monolingual E2E model to the best multilingual E2E model of the Lhasa dialect, the system's performance increased by 8.4% in CER.

Keywords: Transfer learning, End-to-end, Multilingual speech recognition, Low-resource language, Lhasa dialect

1 Introduction

The number of existing languages globally is approximately 7000, and most automatic speech recognition (ASR) efforts deal with languages for which large corpora are readily available, such as Mandarin, English, and French. However, many underresourced languages, such as Tibetan, lack speech data for training ASR systems due to the small population of speakers. Currently, the culture of Tibet is going through radical modernization transformations. Thus, protecting its cultural diversity warrants further attention. The Tibetan language, as the carrier of its culture, should be preserved, and people have attached more importance to the technical contributions of Tibetans. In the Tibetan language family, the Lhasa Tibetan, Khams Tibetan, and Amdi Tibetan are the dominant dialects. The Lhasa dialect spoken in the most populated region of central Tibet has a large canon of Tibetan manuscripts over its long history. Hence, it becomes apparent that applying natural language processing and ASR techniques significantly contributes to preserving the Tibetan language.

Traditional ASR systems require an acoustic model (AM), a language model (LM), and a pronunciation

*Correspondence: longbiao_wang@tju.edu.cn; sheng.li@nict.go.jp
[1]Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
Full list of author information is available at the end of the article

dictionary. In the 1980s, ASR research concentrated on the statistical modeling framework based on the hidden Markov model (HMM) [1]. As is well known, a realistic speech signal is inherently highly variable (due to variations in pronunciation and accent). Therefore, the cardinal form of the HMM is a statistical model that uses a Markov chain to represent the linguistic structure. Meanwhile, it also uses a set of probability distributions to account for the variability in the acoustic realization utterances [2]. With the emergence of artificial neural networks (ANNs), the research for ASR has centered on integrating neural networks with the essential structure of a hidden Markov model to take advantage of the temporal handling capability of the HMM. In past decades, deep neural networks (DNNs) have advanced AM remarkably [3–5].

Even though DNN-based acoustic models have obtained significant improvement on ASR systems [6–8], the limitations resulting from insufficient resources are obvious. Because of relatively scarce resources and the low number of speakers, the Lhasa dialect does not yet have a mature acoustic corpus for public access. Accordingly, research on Tibetan ASR has previously concentrated on selecting acoustic modeling units [9], incorporating effective tonal information [10] and improving Tibetan ASR systems based on lattice-free maximum mutual information (LFMMI) [11], transfer learning [12], and variational modeling units [10]. Due to the abovementioned resource limitations, the development of Tibetan ASR systems has come to a halt, which has created an urgent need for novel methods.

In recent years, end-to-end (E2E) neural networks have emerged and been applied to ASR tasks [13–17]. The E2E ASR network directly recognizes speech representations into text without a lexicon since it handles AM and LM in a single network without expert knowledge of languages. Generally, it is a simple and straightforward method for directly obtaining excellent recognition results. The sequence labeling problem between variable-length speech frame inputs and label outputs (e.g., phone, character, syllable, word, etc.) has been solved to achieve promising results on ASR tasks. Furthermore, the E2E network offers a broader choice of modeling units. Different types of E2E models have been proposed, i.e., connectionist temporal classification (CTC) [18, 19], attention-based encoder-decoder E2E [20, 21], E2E LFMMI [22], and joint CTC and attention E2E models (CTC/attention) [23–26].

More recently, the E2E transformer model [27] was proposed to address neural machine translation and applied to ASR tasks [28–31] and achieved superior performance in certain tasks. Researchers further applied transformer-based models to deal with low-resource languages [32, 33]. However, their work only focused on multilingual training without language-specific training methods, especially for Tibetan. Transfer learning, first proposed in the low-resource machine translation field [34], has been used to improve the low-resource ASR performance by initializing with high-resource languages [35, 36]. Furthermore, the multilingual training method also improved low-resource ASR tasks, thus allowing the model to learn the information across languages [37]. However, the out-of-vocabulary (OOV) problem is caused by the limited training set, and given that the E2E ASR models are always data-hungry, this remains a challenge for low-resource ASR tasks.

In our previous work [38], highly compressed modeling units (Tibetan morphemic radicals) were used to solve the OOV problem, which proved to be effective in experiments. The present work further investigates the initialization strategy with different languages and proposes a novel multilingual transformer-based ASR system for the Lhasa dialect. We provide more detailed background knowledge and explain the technology descriptions as follows. First, the ASR model is trained with different source languages closely related to Tibetan to evaluate the positive effects of different source languages on the Tibetan ASR model. An effective method for this is to select a proper well-resourced language as a source language or joint-training language. Second, a novel Lhasa dialect ASR system is proposed to be initialized by a well-resourced language. It is then fine-tuned with multilingual training by four joint-training languages and multilevel modeling units (characters and radicals in Tibetan). In the training period, different modeling units are regarded as separate languages. The low-resource problem can be solved to a certain extent using this strategy.

The rest of this paper is organized as follows. The related studies are reviewed in Section 2. In Section 3, our unique optimization method, which is proposed for the first time in this paper, is introduced. In Section 4, the task data are evaluated, and the baseline systems are trained. In Section 5, we use the proposed methods to improve the end-to-end ASR system for the Lhasa dialect. In Section 6, we conclude.

2 Related works

The models and techniques most related to this paper are summarized as follows.

2.1 End-to-end transformer model

The architecture of the ASR transformer stacks multi-head attention (MHA) and positionwise fully connected layers for both the encoder and the decode. Each transformer encoder and decoder is a stack of N blocks. The lth block in the decoder maps the input sequence $X = \{x_1 ..., x_n\}$ to the continuous representations $Z = \{z_1 ..., z_n\}$. Given Z, the decoder generates the output sequence $Y =$

$\{y_1..., y_n\}$. The detailed radical roots for the transformer are described as follows.

Since the transformer model relies on a self-attention mechanism with no recurrence, the model cannot handle the sequential order of the inputs. For this reason, positional encodings are applied to the input token embeddings to provide positional information in the model.

2.1.1 Positional encoding

Since the transformer model relies on a self-attention mechanism with no recurrence, the model cannot handle the sequential order of the inputs. For this reason, positional encodings are applied to the input token embeddings to provide positional information in the model.

$$X_i^l = emb_t[w_i] + emb_p[i], \tag{1}$$

where w_i is the ith input token, X^l is the input sequence of the lth block, and emb_t and emb_p denote a learned token embedding matrix and a learned positional embedding matrix, respectively.

2.1.2 Multihead self-attention

The attention function can be described as mapping a query to an output with a set of key-value pairs. The output is a weighted sum of the values. We denote queries, keys, and values as Q, K, and V, respectively. Following the original implementation [27], scaled dot-product attention is employed as the attention function. Hence, the output can be calculated as

$$\mathrm{A}(Q, K, V) = \mathrm{S}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2}$$

where A denotes the attention function, S is the softmax function, and d_k is the dimension of key vectors.

The purpose of multihead attention is to compute multiple independent attention heads in parallel and then concatenate the results and project again. The multihead self-attention in the lth block can be calculated as

$$\mathrm{MHD}(X^l) = \mathrm{Concat}(h_1.., h_r)W^O, \tag{3}$$

$$h_i = \mathrm{A}(X^l W_i^Q, X^l W_i^K, X^l W_i^V), \tag{4}$$

where MHD denotes multihead self-attention, X^l is the input sequence of the lth block, r is the number of heads, and W_i^Q, W_i^K, W_i^V, and W^O are parameter matrices.

2.1.3 Positionwise feedforward layer

The second sublayer in a block is the positionwise feedforward layer, which is applied to each position separately and independently. The output of this layer can be calculated as

$$\mathrm{FFN}(x) = max(0, x \cdot W_1 + b_1)W_2 + b_2, \tag{5}$$

where FFN denotes the feedforward layer, W_1 and W_2 are parameter matrices, and b_1 and b_2 are parameter biases. The *max* function is used to compare the value of $x \cdot W_1 + b_1$ with the 0 vector and outputs a larger value.

2.1.4 Residual connection and layer normalization

The residual connection is added around the two sublayers followed by layer normalization. The output of the lth block can be calculated as

$$H^l = \mathrm{LN}(\mathrm{MHD}(X^l) + X^l), \tag{6}$$

$$X^{l+1} = \mathrm{LN}(\mathrm{FFN}(H^l) + H^l). \tag{7}$$

where LN denotes layer normalization, H^l is the output of layer normalization, and X^{l+1} is the output of the lth block and the input of the $l+1$th block.

2.2 Transformer-based end-to-end ASR systems

2.2.1 Monolingual ASR tasks

The transformer-based model [27] is a known solution that improves various ASR tasks [28, 29, 31]. To this end, speech features are transformed and normalized into an appropriate dimension for inputting to the model. The transformer model for the machine translation task can be applied to speech recognition tasks. A significant difference from the standard E2E model [20, 21] is that the transformer-based acoustic model relies on nonrecurrence radicals [27], multihead self-attention (MHA), positional encoding (PE), and positionwise feedforward networks (PFFN), as mentioned in Section 2.1.

The ASR-transformer encoder maps an input sequence to a sequence of intermediate representations as to the input to the ASR-transformer decoder, which generates an output sequence of symbols (e.g., phonemes, syllables, words, subwords, or words). A monolingual model chooses different modeling units, such as phonemes, morphemes, words, and subwords [39]. In contrast, the transformer model is powerful for learning the mappings between acoustic features and sentences in the training period and adopting the knowledge to recognize unseen acoustic features in the decoding process. It has made significant progress on the public corpus and revealed the powerful advantages of the multihead self-attention mechanism.

2.2.2 Multilingual ASR tasks

The multilingual transformer resembles previous monolingual transformer models in that both are a stack of multilayer encoder-decoder units that use the multihead self-attention mechanism and position feedforward network to model the acoustic feature sequences. The softmax layer in the decoder is the only distinction between the two models. In the monolingual transformer model, the final output node is monolingual, while

in the multilingual counterpart, the final output node is multilingual with mixed modeling units (Tibetan and Chinese characters, for example) of multiple languages. While the multilingual DNN model has different softmax layers for different languages, the multilingual transformer model has a single softmax layer without language identification.

Generally, the transformer can choose multiple modeling units. This idea originates from the general phone set. It also has no requirement for the consistency of different languages' modeling units, which means it has little dependence on expert knowledge. Taking Chinese and English as examples, it is feasible to jointly train Chinese characters and English words when modeling similar languages. The system is improved for performance and robustness by using the subwords as modeling units.

2.3 Background knowledge of Lhasa Tibetan language

As we introduced in Section 1, the Tibetan language belongs to the Sino-Tibetan family and includes three dialects: Lhasa Tibetan, Khams Tibetan, and Amdo Tibetan. The geographical distribution is as shown in Fig. 1.

As shown in Fig. 2, a typical Lhasa Tibetan character has a set of essential radicals root script (Root.), prescript (Pre.), superscript (Super.), subscript (Sub.), vowels (Vo.), and postscript (Post.) to express a wide range of grammatical categories and speech changes (e.g., number, tense and case), thus resulting in an extensive vocabulary. Thus, ASR performance can be affected by the phone set defined by different combinations of these radicals. The number of actual initials in the Lhasa dialect is 28, while Tibetan finals depend on the possible combinations of vowels and character postscripts.

3 Proposed method for modeling low-resource Tibetan dialect

In this section, our novel modeling method is introduced in detail.

3.1 Tibetan radical modeling unit

The rules for assembling and disassembling Tibetan characters and radicals are shown in Fig. 3. A Tibetan character is further segmented into a sequence of subcharacter tokens. The vertically stacking radicals (superscript, ROOT script, subscript, and vowels) in a character are separated and treated as individual units. A boundary marker <-> is used between two consecutive characters. Linguists have confirmed that the original characters can be recovered quickly with the existing boundary marker <-> and radicals. Thus, the set of subcharacter units called basic-57 consists of 56 Tibetan radicals and a boundary marker [38]. Of course, some languages have a similar structure to Tibetan characters and can be disassembled and combined. Therefore, the idea of creating a coarse-grained modeling unit can also be applied to languages with such characteristics, but it is not a standard method in any language.

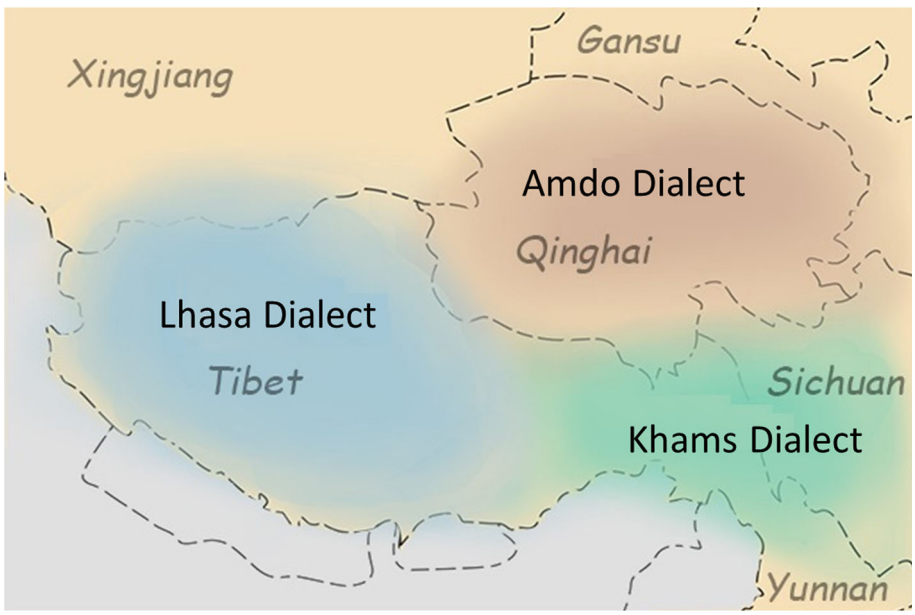
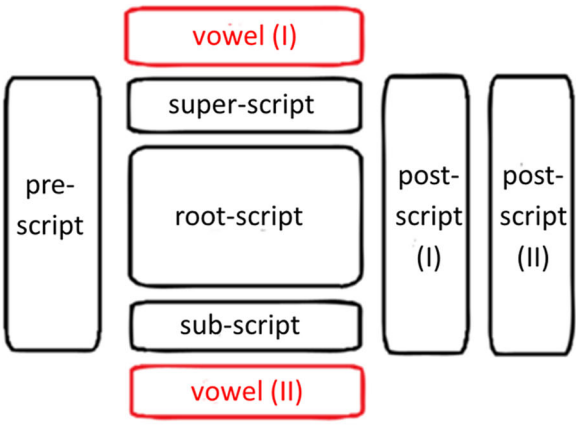


Fig. 1 Three major Tibetan dialects



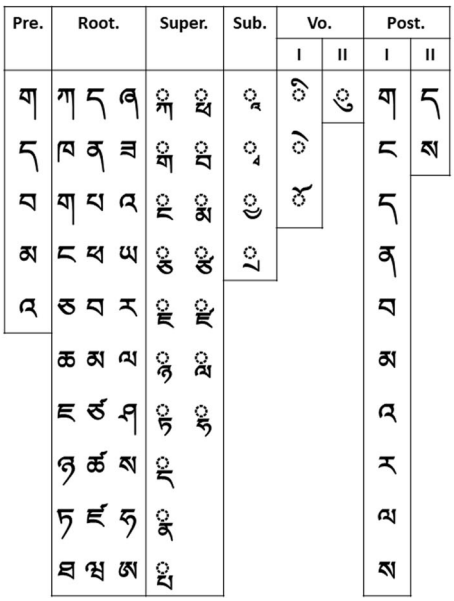


Fig. 2 The structure of a character in the Tibetan writing systems

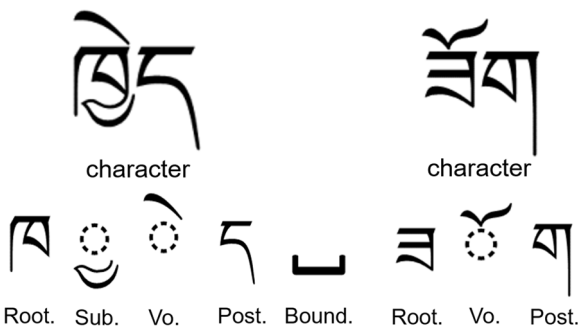


Fig. 3 Segmenting characters to subcharacter units

Language family	Languages
Indo-European	English, Spanish, French, Italian, **Sinhalese, Bengali, Nepali**
Sino-Tibetan	**Mandarin**, Myanmar, **Tibetan**
Austric	Vietnamese, Thai, Indonesian
Others	Japanese, Korean

Fig. 4 A brief summary of language family

3.1.1 Monolingual baseline systems

In this study, Tibetan characters are primarily used as a coarse-grained modeling unit to build a character-level baseline system on the E2E transformer architecture. However, due to a lack of resources, modeling with character-level granularity may result in sparse data. According to the composition of Tibetan characters, we further choose Tibetan radicals (basic-57) as modeling units to build a radical-level baseline system. After significantly compressing the word-level modeling units, the number of modeling units is reduced by two orders of magnitude to alleviate sparse training data on the small-scale training set.

3.2 Proposed transfer learning strategies for low-resource languages

There are many strategies to solve the problem of data sparsity. To this end, two typical methods were employed and then combined with selected source languages in this study.

3.2.1 Initialization strategy

The languages of the world have many differences in pronunciation, word formation, and grammar, but some languages have certain similarities. The main criterion for evaluating the similarity among languages is the classification of language families. It is natural to believe that using languages similar to the Lhasa Tibetan dialect, especially those in the same language family (e.g., Mandarin in Fig. 4), would lead to a well-trained ASR model to efficiently initialize the Lhasa dialect ASR model. Therefore, we chose several resource-rich languages in the same language family as the source languages to pretrain the model for our Tibetan ASR task. After pretraining, the source language E2E model with optimal performance was selected as an initialization model.

In addition, three relatively widespread languages (Bengali, Nepali, and Sinhalese) in Southern Asia were included from OpenSLR[1] as the basis for comparative experiments. Bengali is the official language of

[1] http://www.openslr.org

Bangladesh, West Bengal, and the Tripura states in India, which comprise approximately 270 million people. Nepali is spoken in Nepal, Bhutan, and some regions of India. It is the official language of Nepal, which has a population of approximately 16 million speakers. Sinhalese is the primary, official language of Sri-Lanka and has more than 13 million speakers. Although these languages belong to another language family, they have the same character structure as Tibetan, and all four languages are deeply affected by the ancient Sanskrit language.

This optimization strategy is specifically designed for low-resource speech recognition tasks on the transformer. It can compensate for the data-hunger problem of end-to-end models by sharing the parameters of the resource-rich speech recognition model. In this paper, we compare the contribution of different source languages to the Lhasa dialect ASR model.

3.2.2 Multilingual training

Our multilingual system is based on two types of modeling units and several highly related and resource-rich languages to jointly train the initialization model. The transcriptions on the resources were labeled with different language tags. The two different modeling units also worked as two different languages, similar to those operating in the self-fusion system. This system performs speech recognition and language identification; hence, it improves the accuracy of Lhasa dialect speech recognition by incorporating information across languages.

For multilingual training, several ASR models with different source languages as initialization models are built first and fine-tuned with the Lhasa dialect training set (Lhasa-TRN) to compare the effectiveness of using different source languages. Second, a novel Lhasa dialect ASR system was initialized by a resource-rich language, and then fine-tuned for multilingual training by using four joint-training languages and multilevel modeling units.

4 Task description and baseline systems

In this section, we will describe our dataset and experimental settings for baseline systems.

4.1 Datasets and the DNN-HMM ASR system for Lhasa dialect

The Lhasa speech corpus contains 35.82-hour speech data corresponding to more than 38,700 sentences collected from 13 male and 10 female native Lhasa Tibetan speakers. The recording script is mainly composed of declarative sentences covering a wide range of topics. The speech signal is sampled at 16 kHz with 16-bit quantization. Table 1 summarizes the training set (Lhasa-TRN), development set (Lhasa-DEV), and testing set (Lhasa-TST).

The pronunciation dictionary is provided by the Institute of Ethnology and Anthropology of the Chinese Academy of Social Sciences. The dictionary uses the rules for combinations of initials and vowels containing 29 initials and 48 finals. The dictionary has 2100 entries and covers all Tibetan characters appearing in the Tibetan Lhasa dialect database. This set of pronunciation dictionaries will be used to construct the decoder in the experiment to build a hybrid speech recognition system. The E2E framework does not rely on this pronunciation dictionary.

Table 1 Speech corpus of Lhasa dialect

Datasets	#Speakers	#Utterances	Hours
Training (**Lhasa-TRN**)	10M + 7F	36,090	31.9
Development (**Lhasa-DEV**)	3M + 3F	1,700	1.5
Testing (**Lhasa-TST**)	3M + 3F	2,664	2.4

The training data for the language model used in this paper contain two parts: Tibetan text data obtained from Wikipedia and teaching materials from middle schools in five Tibetan provinces. In total, there are 14,430 Tibetan sentences. The language model uses a 3-gram model and the Kneser-Ney smoothing method. This language model is also not used for transformer modeling in this paper. We use the same experimental settings as [38] to build our DNN-HMM ASR system. The ASR performance of their system was 35.9% of CER%.

4.2 The monolingual end-to-end ASR baseline systems for Lhasa dialect

In this section, we build two monolingual E2E transformer speech recognition systems using the Lhasa dialect only. Compared with the hybrid speech recognition framework, the dataset of the transformer framework is the same as that of the hybrid framework.

As mentioned above, the two modeling units are not related to the pronunciation dictionary used to model the Lhasa dialect. For the character-level modeling unit, a total of 2072 Tibetan characters were obtained from the transcriptions of audio data in the training set. The subcharacter unit set is basic-57, consisting of 56 Tibetan radicals and a boundary marker, as introduced in Section 3.1.

Four additional tags are added to each modeling unit table, namely, OOV tags (UNK), fill tags (PAD), start tags (SENT), and end tags (SENT) to accommodate the transformer model. Since the transformer-based ASR is a sequence-level task, the former two types of tags are always used to represent the out-of-vocabulary issue and used to fill the shorter sentence. In contrast, the latter two represented the starting and ending of a sentence during the decoding stage. Therefore, there are 2076 character-level Lhasa dialect modeling units and 61 radical-level Lhasa dialect modeling units used to build monolingual transformer-based speech recognition systems based

Table 2 Major experimental settings

Model structure			
Attention-heads	8	Decoder-blocks	6
Hidden-units	512	Residual-drop	0.3
Encoder-blocks	6	Attention-drop	0.0
Training settings			
Max-length	5000	GPUs (K40m)	4
Tokens/batch	10000	Warmup-steps	12000
Epochs	30	Steps	300000
Label-smooth	0.1	Optimizer	Adam
Testing settings			
Ave. chkpoints	Last 20	Batch-size	100
Length-penalty	0.6	Beam-size	13
Max-length	50	GPUs (K40m)	4

on random initialization. All experiments are based on the implementation of transformer-based neural machine translation (NMT) [27] in tensor2tensor [2]. The training and testing settings are similar to [31] and listed in Table 2.

The experiment uses 40-dimensional Fbank features to characterize the original audio data, with a window length of 25 ms and a frameshift of 10 ms. Conventional operations, such as CMVN, are carried out with first-order and second-order difference calculations. To adapt to the transformer model, referring to the feature processing method [40], first stitch the current frame and the 3 adjacent frames on its left side and then downsample 3 frames to prevent feature redundancy. Therefore, the actual acoustic feature dimension is 480. Feature extraction experiments are also performed using the Kaldi toolbox.

The Tibetan character- and radical-level modeling units randomly initialize all model parameter settings with the 31.9-hour Lhasa dialect training set. In the testing period, the speech sequences from the test set (Lhasa-TST) are decoded, and the character error rate (CER%) is used to evaluate our models. When using Tibetan radical modeling units, combination postprocessing must restore it to a Tibetan sequence and calculate the CER. The decoded sequence is a series of Tibetan radical sequences containing word boundary markers. The following radical-based experiments are processed in this way.

In Table 3, Char. with no pretraining is an E2E model that selects Tibetan characters as modeling units, while Subchar. with no pretraining selects Tibetan radicals as modeling units. The performances of the two models trained with a random initialization were rather poor (97.94% and 58.63%, respectively), probably because of relatively scarce training data. In contrast, the parameters of the transformer-based acoustic model are relatively large (more than 200 M). In the next sections, the other proposed methods will be introduced to maximize the use of our limited data.

[2] https://github.com/tensorflow/tensor2tensor

Table 3 Performance of Lhasa dialect ASR systems with four source languages as pretrained models

Pretrained model	Char. (%)	Subchar. (%)	Ave. CER (%)
No pretraining	97.94	58.63	78.29
Aishell-1 train	37.19	34.31	35.75
Aishell-1 train_sp	35.95	33.64	34.80
Sinhala train	35.81	35.37	35.59
Sinhala train_sp	35.66	35.56	35.61
Bengali train	38.16	35.81	36.99
Bengali train_sp	35.87	34.79	35.33
Nepali train	36.40	35.14	35.77
Nepali train_sp	38.02	37.14	37.58

The ASR model initialized with Aishell-1 train_sp significantly (the two-tailed *t* test at p value < 0.05) outperforms other models

5 The improved end-to-end ASR systems for Lhasa dialect

In this section, our new method is introduced to improve E2E ASR systems based on the three proposed methods.

5.1 Effective model initialization schemes

Based on our proposed initialization method, a language similar to Tibetan is selected from the language family to build a well-trained transformer model as the initialization model and to compensate for the resource-poor training data. The original softmax layer is replaced with the language-specific and randomly initialized softmax layer. In this paper, a well-trained transformer-based ASR model (8 head-attention, 6 encoder-blocks and 6 decoder-blocks with 512 nodes) with a CER of 9.0% is regarded as the initialization model. This model is trained using 178 hours of Mandarin speech data selected from the Aishell dataset [41]. We also select three relatively resource-rich languages (Bengali, Nepali, and Sinhalese) similar to Tibetan to construct the initialization models mentioned in Section 3.2. A speed perturbation is utilized to augment the data three times. The specific duration is shown in Table 4.

The Aishell-1 database was trained on the training set (train) and the training set (train_sp) with triple speed

Table 4 Comparison of the duration of the four source languages

Duration (hours)	Aishell-1	Sinhala	Bengali	Nepali
train	150.9	214.6	214.5	153.6
train_sp	455.6	648.2	647.7	464.0

perturbation with word error rates of 8.74% and 7.46%, respectively. The trainings on the Nepali database using the same sets resulted in 17.80% and 14.49% word error rates (WERs), respectively; on the Bengali database, they obtained 19.70% and 16.04% WERs, respectively; and on the Sinhala database, they obtained 23.76% and 18.60% WERs, respectively. The parameters of each source language model are transferred to the speech recognition of the Lhasa dialect. The contribution of the selected transformer models trained by the four source languages to the monolingual ASR system is shown in Table 3.

In Table 3, Aishell-1 train_sp significantly outperforms other models (i.e., the two-tailed t test at p value < 0.05). The Tibetan subcharacter-based modeling units perform better than the Tibetan character-level modeling units. The Tibetan subcharacter-level modeling units obtained the best performance at 33.64% CER with the Aishell-1 train_sp data, which significantly exceeded the baseline system performance on the hybrid speech recognition framework. Hence, using a highly relevant language, especially in the same language family, as a source language effectively initializes the target transformer model.

5.2 The self-fusion end-to-end ASR system for the Lhasa dialect

In this section, the system will be self-fused by training it using two different levels of modeling units, which are regarded as two languages. This method was proposed in our previous work [38], but the model was initialized by 178-h speech data of Aishell-1 in [38]. It is worth mentioning that train_sp of the Aishell-1 database is used, as shown in Table 4, to initialize the transformer model, which is the best initialization method shown in Section 5.1.

In our experiment, the transformer model was trained with basic-57 and Char. 2072 together based on the multilingual training method. To distinguish between the two modeling units, labels were created for each modeling unit as tib_char and tib_radical. This self-fusion model (Multiunit transformer) significantly improved the system performance of monolingual ASR baseline systems. The postprocessing for a decoded radical sequence is used as introduced in Section 4.2. A comparison of the performance of the different systems is shown in Table 5. The self-fusion ASR system's performance with a CER of 32.99% is obviously better than baseline systems, which have an average CER of 78.29%, and better than the best monolingual ASR systems based on characters or subcharacters, which have an average CER of 34.80%, as shown in Table 5. The self-fusion ASR model is also better than the DNN-HMM-based ASR model.

The experimental results show that the different modeling units are complementary in performance. The E2E transformer model of the Lhasa dialect can be further improved based on multilingual speech recognition to fuse two monolingual recognition systems.

Table 5 Performance of different Lhasa dialect modeling units, self-fusion systems (pretrained by Aishell-1 train_sp) and multilingual speech recognition systems with multi-level modeling units (pretrained by different languages)

Pretrained model	Transformer	CER or Ave. CER (%)
No pretraining	Char. or Subchar.	78.29
Aishell-1 train_sp	Char. or Subchar.	34.80
Aishell-1 train_sp	Self-fusion	32.99
No pretraining	Multilingual	33.18
Aishell-1 train_sp	Multilingual	30.79
Sinhala train	Multilingual	31.93
Bengali train_sp	Multilingual	31.82
Nepali train	Multilingual	31.66

The multilingual-ASR model initialized with Aishell-1 train_sp significantly (i.e., the two-tailed t test at p value < 0.05) outperforms other models

5.3 Lhasa dialect multilingual speech recognition system

There are four resource-rich languages and two different modeling units, which are regarded as two languages, to jointly train a Lhasa dialect multilingual speech recognition system based on the five initialization models introduced above. This system can handle language identification and speech recognition tasks. The modeling units of the multilingual system are composed of Mandarin, Tibetan characters and radicals, and the word-level units of Bengali, Sinhalese, and Nepali. To develop the ability to identify languages, we marked the languages with different tags. Therefore, there are 6703 modeling units in the full model.

Similar to the Lhasa dialect's self-fusion system, all transcriptions must be marked with the corresponding language tags at the front. To better connect with the existing basic experiments, we select several initialization models with excellent performance in the monolingual speech recognition task.

In Table 5, from top to bottom, a comparison is made on the CERs obtained by training transformer models using different initialization models for multilingual speech recognition. It is found that a reasonable initialization model can still obtain a performance gain even when training with resource-rich languages. The best initialization method has a relative improvement of 7.2% compared to the case without initialization and significantly (the two-tailed t test at p value < 0.05) outperforms other models. Horizontally, on the table, a comparison is shown for the best Lhasa dialects with monolingual, self-fusion, and multilingual speech recognition systems. Their CERs were 33.64%, 32.99%, and 30.79%, respectively, and their initialization models were consistent.

According to this set of strictly controlled experiments, for low-resource speech recognition tasks, it is clear that joint training with the correlated source languages improves the performance of low-resource speech recognition. The system's performance increased by 8.4% in CER from the best monolingual model of the Lhasa dialect to the best multilingual model.

6 Conclusion and future work

In this paper, we focused on training transformer-based E2E ASR systems for the Lhasa dialect. We investigated a compressed acoustic modeling unit set, effective initialization strategies, multiunit training, and multilingual speech recognition for low-resource data to solve the issue of low-resource data. In the monolingual E2E speech recognition system, we achieved a relative 6.3% gain in CER performance compared to hybrid speech recognition. From the best monolingual model of the Lhasa dialect to the best multilingual E2E model, the system's performance increased by 8.4% in CER. Experiments show that our proposed methods effectively model the low-resource Lhasa dialect and outperform the conventional DNN-HMM baselines and E2E baseline systems. Thus, this study provides a new direction for research on low-resource languages.

In future work, we will try a larger transformer structure to investigate the function of the model structure. The correlation between the source and target languages is worth discussing to obtain promising performance. Furthermore, we will deeply connect language identification with speech recognition tasks to probe whether more low-resource languages with only language labels can further improve the performance.

Abbreviations

ASR: Automatic speech recognition; AM: Acoustic model; LM: Language model; HMM: Hidden Markov model; ANN: Artificial neural networks; DNN: Deep neural network; LFMMI: Lattice-free maximum mutual information; E2E: End-to-end; CTC: Connectionist temporal classification; MHA: Multihead self-attention (MHA); PE: Positional encoding; PFFN: Positionwise feedforward networks; WER: Word-level error rate; OOV: Out of vocabulary; NMT: Neural machine translation

Acknowledgements

The authors thank Dr. Hongcui Wang for building concrete baseline systems and Mr. Kuntharrgyal Khyuru for discussions about Tibetan language and text processing.

This work was supported by the National Key R&D Program of China (Grant NO. 2018YFB1305200), by the National Natural Science Foundation of China (Grant NO. 61771333), and by the Tianjin Municipal Science and Technology Project (Grant NO. 18ZXZNGX00330).

Authors' contributions

LW and JD participated in database construction and supplemented the details of the proposed methods. SL participated in the experimental design and analyzed the experimental results. SQ completed the proposed methods of this paper, participated in experimental realization, and was a major contributor in writing the manuscript. LP processed the data and participated in the experimental realization. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Key R and D Program of China under Grant 2018YFB1305200, the National Natural Science Foundation of China under Grant 61771333, and the Tianjin Municipal Science and Technology Project under Grant 18ZXZNGX00330.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China. [2]National Institute of Information and Communications Technology (NICT), Kyoto, Japan. [3]Japan Advanced Institute of Science and Technology, Ishikawa, Japan. [4]Huiyan Technology(Tianjin) Co., Ltd., Tianjin 300384, China.

Received: 26 January 2021 Accepted: 28 November 2021
Published online: 12 January 2022

References

1. B. H. Juang, L. R. Rabiner, Hidden Markov models for speech recognition. Technometrics. **33**(3), 251–272 (1991)
2. B.-H. Juang, L. R. Rabiner, *Automatic speech recognition–a brief history of the technology development, vol. 1*. (Georgia Institute of Technology. Atlanta Rutgers University and the University of California, Santa Barbara, 2005), p. 67
3. L. Deng, G. Hinton, B. Kingsbury, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. New types of deep neural network learning for speech recognition and related applications: an overview (IEEE, 2013), pp. 8599–8603
4. G. E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 30–42 (2011)
5. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al*, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Proc. Mag. **29**(6), 82–97 (2012)
6. O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(10), 1533–1545 (2014)
7. H. Sak, A. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128 (2014)
8. A. Graves, N. Jaitly, A.-r. Mohamed, in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. Hybrid speech recognition with deep bidirectional LSTM (IEEE, 2013), pp. 273–278
9. H. Wang, K. Khyuru, J. Li, G. Li, J. Dang, L. Huang, in *Proc. APSIPA ASC*. Investigation on acoustic modeling with different phoneme set for continuous Lhasa Tibetan recognition based on DNN method, (2016)
10. J. Li, H. Wang, L. Wang, J. Dang, K. Khuru, G. Lobsang, in *Proc. ISCSLP*. Exploring tonal information for lhasa dialect acoustic modeling, (2016)
11. D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, S. Khudanpur, in *Proc. INTERSPEECH*. Purely sequence-trained neural networks for ASR based on lattice-free MMI, (2016)
12. J. Yan, Z. Lv, S. Huang, H. Yu, in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*. Low-resource Tibetan dialect acoustic modeling based on transfer learning, (2018)
13. J. Chorowski, D. Bahdanau, K. Cho, Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent nn: first results. arXiv preprint arXiv:1412.1602 (2014)
14. W. Chan, N. Jaitly, Q. Le, O. Vinyals, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Listen, attend and spell: a neural network for large vocabulary conversational speech recognition (IEEE, 2016), pp. 4960–4964

15. S. Watanabe, T. Hori, S. Kim, J. R. Hershey, T. Hayashi, Hybrid ctc/attention architecture for end-to-end speech recognition. IEEE J. Sel. Top. Sig. Process. **11**(8), 1240–1253 (2017)
16. A. Graves, A.-r. Mohamed, G. Hinton, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Speech recognition with deep recurrent neural networks (IEEE, 2013), pp. 6645–6649
17. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, in *Advances in Neural Information Processing Systems*. Attention is all you need, (2017), pp. 5998–6008
18. A. Graves, N. Jaitly, in *Proc. ICML*. Towards end-to-end speech recognition with recurrent neural networks, (2014)
19. Y. Miao, M. Gowayyed, F. Metze, in *Proc. IEEE-ASRU*. EESEN: end-to-end speech recognition using deep RNN models and WFST-based decoding, (2015), pp. 167–174
20. J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, in *Proc. NIPS*. Attention-based models for speech recognition, (2015)
21. W. Chan, N. Jaitly, Q. Le, O. Vinyals, in *Proc. IEEE-ICASSP*. Listen, attend and spell: a neural network for large vocabulary conversational speech recognition, (2016)
22. H. Hadian, H. Sameti, D. Povey, S. Khudanpur, in *Proc. INTERSPEECH*. End-to-end speech recognition using lattice-free MMI, (2018)
23. S. Watanabe, T. Hori, S. Kim, J. R. Hershey, T. Hayashi, Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE J. Sel. Top. Sig. Process. **11**(8), 1240–1253 (2017)
24. S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique, Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, T. Ochiai, in *Proc. INTERSPEECH*. Espnet: end-to-end speech processing toolkit, (2018)
25. S. Ueno, H. Inaguma, M. Mimura, T. Kawahara, in *Proc. IEEE-ICASSP*. Acoustic-to-word attention-based model complemented with character-level CTC-based model, (2018), pp. 5804–5808
26. T. Hori, S. Watanabe, Y. Zhang, W. Chan, in *Proc. INTERSPEECH*. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM, (2017)
27. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, in *arXiv Preprint Arxiv:1706.03762*. Attention is all you need, (2017)
28. L. Dong, S. Xu, B. Xu, in *Proc. IEEE-ICASSP*. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition, (2018)
29. S. Zhou, S. Xu, B. Xu, in *arXiv Preprint Arxiv:1806.05059*. Multilingual end-to-end speech recognition with a single transformer on low-resource languages, (2018)
30. S. Zhou, L. Dong, S. Xu, B. Xu, in *arXiv Preprint Arxiv:1805.06239*. A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on Mandarin Chinese, (2018)
31. S. Zhou, L. Dong, S. Xu, B. Xu, in *Proc. INTERSPEECH*. Syllable-based sequence-to-sequence speech recognition with the transformer in Mandarin Chinese, (2018)
32. V. M. Shetty, M. Sagaya Mary N.J., in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Improving the performance of transformer based low resource speech recognition for Indian languages, (2020), pp. 8279–8283. https://doi.org/10.1109/ICASSP40776.2020.9053808
33. S. Zhou, S. Xu, B. Xu, Multilingual end-to-end speech recognition with a single transformer on low-resource languages. arXiv preprint arXiv:1806.05059 (2018)
34. B. Zoph, D. Yuret, J. May, K. Knight, Transfer learning for low-resource neural machine translation. arXiv preprint arXiv:1604.02201 (2016)
35. J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, T. Hori, in *2018 IEEE Spoken Language Technology Workshop (SLT)*. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling (IEEE, 2018), pp. 521–527
36. J. Meyer, *Multi-task and transfer learning in low-resource speech recognition*. (The University of Arizona, 2019)
37. S. Dalmia, R. Sanabria, F. Metze, A. W. Black, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Sequence-based multi-lingual low resource speech recognition (IEEE, 2018), pp. 4909–4913
38. L. Pan, S. Li, L. Wang, J. Dang, in *Proc. APSIPA ASC*. Effective training end-to-end ASR systems for low-resource Lhasa dialect of Tibetan language, (2019)
39. T. N. Sainath, R. Prabhavalka, S. Kumar, S. Lee, A. Kannan, D. Rybach, V. Schogol, P. Nguyen, B. Li, Y. Wu, Z. Chen, C. Chiu, in *Proc. IEEE-ICASSP*. No need for a lexicon? Evaluating the value of the pronunciation lexicon in end-to-end models, (2018), pp. 5859–5863
40. A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, R. Prabhavalkar, in *Proc. IEEE-ICASSP*. An analysis of incorporating an external language model into a sequence-to-sequence model, (2018), pp. 5824–5828
41. H. Bu, J. Du, X. Na, B. Wu, H. Zheng, in *Proc. Oriental COCOSDA*. AIShell-1: an open-source Mandarin speech corpus and a speech recognition baseline, (2017)

Publisher's Note