HIERARCHICAL SOFTMAX FOR END-TO-END LOW-RESOURCE MULTILINGUAL SPEECH RECOGNITION

Qianying Liu1*, Zhuo Gong3*, Zhengdong Yang1*, Yuhang Yang2, Sheng Li4, Chenchen Ding4, Nobuaki Minematsu3, Hao Huang2, Fei Cheng1, Chenhui Chu1, Sadao Kurohashi1

1Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto, Japan
2School of Information Science and Engineering, Xinjiang University, Urumqi, China
3Department of EEIS, Graduate School of Engineering, The University of Tokyo, Tokyo, Japan
4National Institute of Information and Communications Technology (NICT), Kyoto, Japan

ABSTRACT

Low-resource speech recognition has been long-suffering from insufficient training data. In this paper, we propose an approach that leverages neighboring languages to improve low-resource scenario performance, founded on the hypothesis that similar linguistic units in neighboring languages exhibit comparable term frequency distributions, which enables us to construct a Huffman tree for performing multilingual hierarchical Softmax decoding. This hierarchical structure enables cross-lingual knowledge sharing among similar tokens, thereby enhancing low-resource training outcomes. Empirical analyses demonstrate that our method is effective in improving the accuracy and efficiency of low-resource speech recognition.

Index Terms: Speech recognition, Acoustic model, End-to-End Multilingual Model

1. INTRODUCTION

Automatic speech recognition (ASR) systems have gained remarkable progress in the past few years. Nevertheless, the present ASR systems cater to only approximately 100 out of the 7000 spoken languages worldwide. To address this limitation, multilingual models have garnered much attention. These models can learn universal features, transferable from resource-rich to resource-limited languages, and support multiple languages with a single ASR model. Early studies utilized context-dependent deep neural network hidden Markov models [1], which relied on hand-crafted pronunciation lexicons. However, when adapted to low-resource languages, such systems exhibit limitations due to the absence of sufficient modeling techniques. Attention-based end-to-end (E2E) models simplify training and eliminate the dependence on pronunciation lexicons [2, 3, 4]. Recent studies employing E2E models have focused on learning universal representations at the encoding stage, using transfer learning techniques [5, 6, 7, 8, 9, 10], as well as on hierarchical embedding of phonemes, phones, and phonological articulatory attributes [11]. Meanwhile, large pretrained models [12, 13, 14, 15] and multilingual speech corpora [16, 17, 18, 19, 20] have been investigated for learning pretrained representations.

In this paper, we aim to investigate the explicit transfer of cross-language knowledge at the decoding stage, which has been largely unexplored in prior studies that focus on encoder representations. Based on linguistic studies on the presence of similar modeling unit distributions in neighboring languages [21], we propose an efficient method to capture the similarity among these units, such as characters and globalphones, at the decoding stage. Our approach utilizes Huffman coding to automatically capture the similarity among modeling units, relying only on monolingual data. We introduce hierarchical Softmax (H-Softmax) [22], an approximation softmax inspired by binary trees, to model the similarity during decoding. This structure enables similar units to share decoder representations, thus improving model performance, and also breaks down the expensive softmax step into several binary classifications, thereby enhancing model efficiency [23]. We design a vectorization algorithm that can expedite the training and inference procedures, enabling our method to outperform the vanilla softmax on GPUs in terms of efficiency.

With the combination of the two components, our method:

• Automatically captures cross-lingual modeling unit similarity for multilingual ASR.
• Leverages H-Softmax to achieve higher efficiency and reduce computational complexity.

While previous studies has utilized H-Softmax in the neural language model of ASR [24, 25]. However, to the best of our knowledge, no existing work has investigated the direct application of H-Softmax in the acoustic model. Furthermore, our study is the first to explore the potential of H-Softmax for low-resource multilingual ASR.
2. PROPOSED METHOD

In this section, we introduce the proposed method of this paper, as shown in Fig 1. We first use the training text data to determine the Huffman code for each token as described in subsection 2.1. Then we build the ASR model, where an encoder takes in the source speech signals, and the decoder predicts the Huffman code of target text with H-Softmax. At the inference stage, the model predicts a sequence of post-processed Huffman codes to text.

2.1. Huffman Code

Based on the assumption that neighbour languages share similar token distribution, the concept of our proposed method is to generate a representation code for each token via frequency clustering. We first generate the Huffman code of each token. Formally, given the multilingual token vocabulary \( V = \{t_1, t_2, ..., t_N\} \), where \( N \) is the vocabulary size, we maintain the term frequency set \( S_p = \{p_i\}_{i=1}^N \), where the same token in different languages are considered as one token. With \( S_p \), we generate a Huffman tree of \( V = \{t_i\} \) via frequency clustering and further recursively generate the Huffman code by assigning 0 to the left subtree and 1 to the right subtree.

2.2. Model Architecture

For the ASR model, we use conformer [26] as the encoder and transformer [27] as the decoder. For the decoder, we replace the vanilla softmax with H-Softmax.

H-Softmax organizes the output vocabulary into a tree where the leaves are the vocabulary tokens, and the intermediate nodes are latent variables [22]. We use the Huffman tree generated in subsection 2.1 as the tree for H-Softmax that there is a unique path from the root to each token, forming a complete binary tree. We follow [22] and apply the binary tree H-Softmax. The decoding procedure is transformed into predicting one leaf node of the binary tree at each timestep. Each leaf node, which represents a token, could be reached by a path from the root through the inner nodes. Given the transformer output hidden state \( h \) and trainable node representation vectors \( \{r_i\} \), the final possibility of a leaf node \( w_i \) could be represented as:

\[
P(\text{label} = w_i) = \prod_{k=1}^{\text{path}} \left( \sigma(r_k h) \text{ if path}_k = \text{left}, 1 - \sigma(r_k h) \text{ if path}_k = \text{right}. \right)
\]

where \( n_k \) stands for the \( k \)th node on the path from the root to \( w_i \) such as \( n_0, n_1 \) and \( n_2 \) in Fig 1; path \( k \) stands for the branch leading towards \( w_i \) which is left or right, and \( \sigma(\cdot) \) stands for sigmoid function.

2.3. Efficient Implementation of H-Softmax

While decomposing the Softmax to a binary tree H-Softmax reduces the decoding time complexity from \( O(V) \) to \( O(\log(V)) \), and the train time complexity remains \( O(V\log(V)) \). Since previous H-Softmax implementation[23] is on CPU, considering the order of magnitude difference between CPU’s and GPU’s FLOPs, the challenge of improving the efficiency of model training lies in designing an implementation of H-Softmax for GPU training. We propose a vectorization algorithm to accelerate training and decoding procedures on GPU.

![Fig. 1](image1.png)

**Fig. 1.** The flowchart of the proposed method. The blue lines stand for determination relation, and the red line stands for how the data goes through the model.

![Fig. 2](image2.png)

**Fig. 2.** A typical H-Softmax tree structure. Leaf \( w_3 \) has a virtual child with the same probability of aligning each leaf node to the same depth, so it is conceptually possible for path vectorization.

To explain the core idea of our vectorization algorithm, we show a typical H-Softmax tree structure in Fig 2. Then, we can vectorize log-probability calculations from Eq 1 to the following:

\[
\log P(w_1) = \log [\sigma(r_1^h)\sigma(r_2^h)] - \log[1-\sigma(r_1^h)] \\
\log P(w_2) = \left[ \log[\sigma(r_1^h)\sigma(r_2^h)] - \log[1-\sigma(r_1^h)] \right] \\
\log P(w_3) = \left[ \log[\sigma(r_1^h)\sigma(r_2^h)] - \log[1-\sigma(r_1^h)] \right] \\
\sum_{\text{column}} \log [ 1 * \sigma(r_1^h) + 1 * \sigma(r_2^h) + 0 -1 * \sigma(r_1^h) + 0 ] \\
\sum_{\text{column}} \log (\text{Sign} \circ \sigma(p) + \text{Bias})
\]

where \( \text{Sign} \) is a 3 by 2 matrix of \( \sigma \)'s signs, \( \text{Bias} \) is a 3 by 2 matrix of \( \sigma \)'s biases and \( p \) is the result vector of the in-
ner product between node vectors and $h$. After building the Huffman tree, the $\text{Sign}$ and $\text{Bias}$ matrices are fixed. So in the training stage, leaf node log probabilities can be acquired only by vector operations.

For decoding, we only need leaves with the highest probabilities. To directly calculate this objective, different from training, we also develop a path-encoding-based multi-layer beam searching on GPU for H-softmax\(^1\) to retain the time efficiency advantage of time-space complexity $O(\log(V))$ compared to vanilla softmax’s $O(V)$.

### 3. EXPERIMENT EVALUATIONS

In this section, we evaluate the proposed method in two low-resource settings. To examine the effectiveness of our method in more extensive settings, specifically in the same language group and cross-language groups, we first simulate a low resource setting by down-sampling from a speech-to-text multilingual large-scale dataset. To further verify the performance on natural low-resource languages and other token modeling units, we test our method on an extremely low-resourced zero-shot cross-lingual speech-to-phoneme setting. For the high-resource setting, where every token could be fully trained, modeling of neighbors could no longer be useful, and thus our experiments focus on the low-resource setting.

#### 3.1. Data Description

We sample our speech-to-text datasets from Common Voice Corpus 11.0. We selected three different linguistic groups: Romance, Slavic, and Turkic. For each group, we selected five languages from the corpus and constructed training and testing sets with the validated data of each language. With the data size of many existing datasets being around 20–30 hours [28, 29], we downsampled the training data to the extent of 30 hours per language on average to simulate a low-resource setting. As a result, the total size of training data changed from 5492 hours to 450 hours, with the general downsampling ratio $\lambda = 0.082$. Different downsampling ratios are used for different languages to counter the imbalance of data size among languages. For a set of languages $\{L_1, ..., L_m\}$ with their proportion $\{p_1, ..., p_m\}$, the downsampling ratio for language $L_i$ is obtained by $\lambda_i = \frac{p_i^{\alpha-1}}{\sum_{j=1}^m p_j^\alpha} \lambda$, where $\alpha$ is a smooth coefficient that we set to 0.5. The size of testing sets is the lesser of 10% of the validated data and 10,000 utterances.

For speech-to-phoneme datasets, we use UCLA Phonetic Corpus [19]. The Edo-Ancient Tokyo (bin), Kele-Congo (sbc), Malayalam-Malay (mal), Creole-Cape Verde (kez), Klaor-Liberia (klu), Arabic-Tunisian Spoken (aeb), Makasar-South Sulawesi (mak), Finnish-Finland (fin), Abkhaz-North Caucasian (abk) and Aceh-Sumatra (ace) are testing languages and other languages are used for training.

#### Table 1. Speech-to-text datasets statistics.

| Group    | Language     | Training (Hours) | Testing (Hours) |
|----------|--------------|------------------|-----------------|
| Romance  | Catalan (ca) | 76.3             | 15.3            |
|          | Spanish (es) | 37.6             | 14.4            |
|          | French (fr)  | 55.7             | 13.4            |
|          | Italian (it) | 33.4             | 15.0            |
|          | Portuguese (pt) | 20.7           | 11.4            |
| Slavic   | Belarussian (be) | 63.0           | 13.9            |
|          | Czech (cs)   | 42.5             | 5.8             |
|          | Polish (pl)  | 37.8             | 12.3            |
|          | Russian (ru) | 27.3             | 14.5            |
|          | Ukrainian (uk) | 23.4           | 7.2             |
| Turkic   | Bashkir (ba) | 29.6             | 12.6            |
|          | Kyrgyz (ky)  | 11.3             | 3.8             |
|          | Turkish (tr) | 16.9             | 8.4             |
|          | Tatar (tt)   | 10.0             | 3.0             |
|          | Uzbek (uz)   | 18.1             | 10.4            |

#### Table 2. Speech-to-phoneme datasets.

| Language               | Dataset                  | Hours  |
|------------------------|--------------------------|--------|
| UCLA Phonetic Corpus   | Training (87 lang.)      | 1.86   |
|                        | Testing (10 lang.)       | 0.14   |

#### 3.2. Model Training

For acoustic features, the 80-dimensional log-Mel filterbanks (FBANK) are computed with a 25ms window and a 10ms shift. Besides, SpecAugment [30] is applied to 2 frequency masks with maximum frequency mask ($F = 10$) and 2 time masks with maximum time mask ($T = 50$) to alleviate overfitting.

Both H-Softmax and Softmax models are trained using the same network structure. The networks are constructed using WeNet toolkit [31].\(^1\) Two convolution sub-sampling layers with kernel size $3 \times 3$ and stride 2 are used in the front of the encoder. For model parameters, we use 12 conformer layers for the encoder and 6 transformer layers for the decoder.

Adam optimizer is used with a learning rate schedule with 25,000 warm-up steps. The initial learning rate is 0.00005. 100 max epochs for speech-to-text datasets and 80 max epochs for speech-to-phoneme datasets.

#### 3.3. Speech-to-text Recognition Evaluation

We conducted two sets of experiments on speech-to-text datasets: training with languages within one language group and training with languages across language groups. We tokenized the transcriptions at the character level as we verified that it outperforms tokenizing in the sub-word level (such as BPE).\(^2\) As shown in Table 4, when training with languages

---

\(^1\)Our implementation is https://github.com/Derek-Gong/hsoftmax

\(^2\)Preliminary experiments on Slavic language group with traditional Softmax results in a CER of 8.5% for character-level tokenization, 8.9% for
Table 3. PER% on speech-to-phoneme datasets.

| Model   | bin  | sbc  | mal | kea | klu  | aeb  | mak  | fin  | abk  | ace  | Overall |
|---------|------|------|-----|-----|------|------|------|------|------|------|---------|
| Softmax | 70.4 | 87.4 | 98.0| 83.4| 86.2 | 84.1 | 84.5 | 94.2 | 75.0 | 75.5 | 85.2    |
| H-Softmax | **80.8** | **68.9** | **59.6** | **62.9** | **73.7** | **60.9** | **75.2** | **70.9** | **48.6** | **64.7** |

Table 4. CER% on speech-to-text datasets. Models are trained with languages within the same language group.

| Model | ca  | es  | fr  | it  | pt  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 5.5 | 9.4 | 12.1| 8.8 | 9.7 | 9.1     |
| H-Softmax | **5.3** | **8.8** | **11.3** | **8.4** | **10.0** | **8.8** |

| Model | be  | cs  | pl  | ru  | uk  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 5.0 | 5.4 | 9.2 | 11.6| **11.5** | **8.5** |
| H-Softmax | **4.8** | **5.4** | **8.8** | **10.4** | **11.6** | **8.2** |

| Model | ba  | ky  | tr  | tt  | uz  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 16.9| 18.3| 14.0| 10.2| 20.6| 16.0    |
| H-Softmax | **14.4** | **17.2** | **12.8** | **9.1** | **16.3** | **14.0** |

Table 5. CER% on speech-to-text datasets. Models are trained with all the languages across language groups.

| Model | ca  | es  | fr  | it  | pt  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 7.3 | 9.4 | 15.1| 8.4 | 21.8| 12.4    |
| H-Softmax | **5.7** | **9.0** | **11.3** | **7.3** | **15.4** | **9.7** |

| Model | be  | cs  | pl  | ru  | uk  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 5.8 | 5.8 | 8.5 | 10.6| 11.5| 8.4     |
| H-Softmax | **6.1** | **5.9** | **8.2** | **8.5** | **9.8** | **7.7** |

| Model | ba  | ky  | tr  | tt  | uz  | Overall |
|-------|-----|-----|-----|-----|-----|---------|
| Softmax | 11.8| 13.7| 12.3| 7.6 | 13.5| 11.8    |
| H-Softmax | **11.0** | **13.6** | **10.3** | **7.1** | **13.7** | **11.1** |

Table 6. RTF (real-time factor) of the decoding process of sentence length (tokens per sentences) determines the decoding time difference between Softmax and H-softmax. Longer sentences take more time steps for inference, and the time difference between the two models exponentially increases.

| Model   | tr  | pl  | ru  |
|---------|-----|-----|-----|
| Softmax | 32.6 tok/sent | 47.4 tok/sent | 63.1 tok/sent |
| H-Softmax | **0.018** | **0.018** | **0.026** |

3.4. Zero-shot Cross-lingual Speech-to-phoneme Recognition Evaluation

Our proposed model demonstrated superior performance compared to the conventional model (phone error rate, PER%) across all languages on the UCLA Phonetic corpus, as presented in Table 3. Overall, our approach outperformed the softmax baseline by a significant margin of 20.51% PER. On the ‘bin’ language, the performance gap between softmax and H-softmax was 32.33%. These results showcase the effectiveness of our method in automatically constructing universal representations for multilingual ASR and achieving zero-shot cross-lingual phoneme recognition.

3.5. Decoding Speed

We observed decoding acceleration with our proposed H-Softmax model in Table 6. Decomposing the Softmax output layer to a binary tree reduces the complexity of obtaining probability distribution from O(V) to O(log(V)), which leads to improvement in efficiency. The results also show that the sentence length (tokens per sentences) determines the decoding time difference between Softmax and H-softmax. Longer sentences take more time steps for inference, and the time difference between the two models exponentially increases.

4. CONCLUSION

This paper proposes an automatic method to generate cross-lingual representations, which facilitate the acquisition of cross-lingual universal features in neighboring languages. Our approach employs Huffman coding, which utilizes token frequencies (characters, globalphones, etc.) to bridge different languages. Furthermore, we introduce H-Softmax, which improves model performance by enabling similar units that share Huffman binary representations and accelerates decoding compared to traditional Softmax methods. As future work, we aim to design binary codes that incorporate not only shallow frequency terms but also more semantically meaningful features, such as token embeddings.

5. ACKNOWLEDGEMENTS

This study was supported by JST SPRING Grant No.JPMJSP2110 and Li’s JSPS KAKENHI Grant Scientific Research (C).
6. REFERENCES

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context dependent pre-trained deep neural networks for large vocabulary speech recognition,” IEEE Trans. ASLP, vol. 20, no. 1, pp. 30–42, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[3] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in Proc. ICML, PMLR, 2014, pp. 1764–1772.

[4] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio, “End-to-end attention-based large vocabulary speech recognition,” in Proc. IEEE-ICASSP. IEEE, 2016, pp. 4945–4949.

[5] X. Li and et al., “Towards zero-shot learning for automatic phonemic transcription,” in Proc. AAAI, 2020.

[6] C. Zhu and et al., “Multilingual and crosslingual speech recognition using phonological-vector based phone embeddings,” in Proc. ASRU, 2021.

[7] Sheng Li, Chenchun Ding, Xugang Lu, Peng Shen, Tatsuya Kawahara, and Hisashi Kawai, “End-to-end articulatory attribute modeling for low-resource multilingual speech recognition,” in INTERSPEECH, 2019, pp. 2145–2149.

[8] W. Hou and et al., “Exploiting adapters for cross-lingual low-resource speech recognition,” in Proc. INTERSPEECH, 2020.

[9] W. Hou and et al., “Large-scale end-to-end multilingual speech recognition and language identification with multi-task learning,” in Proc. INTERSPEECH, 2020.

[10] D. Liu and et al., “Learning phone recognition from unpaired audio and phone sequences based on generative adversarial network,” 2021.

[11] X. Li, J. Li, F. Metze, and A. Black, “Hierarchical phone recognition with compositional phonetics,” in Proc. INTERSPEECH, 2021, pp. 2461–2465.

[12] A. Baevski and et al., “wav2vec2.0: A framework for self-supervised learning of speech representations,” in Proc. NeurIPS, 2020.

[13] A. Baevski and et al., “Unsupervised speech recognition,” in Proc. NeurIPS, 2021.

[14] Y. Wang and et al., “A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding,” in arXiv preprint arXiv:2111.02735, 2021.

[15] S. Chen and et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” in arXiv preprint arXiv:2110.13900, 2021.

[16] C. Wang and et al., “Covost: A diverse multilingual speech-to-text translation corpus,” in Proc. LREC, 2020.

[17] C. Wang and et al., “Covost 2: A massively multilingual speech-to-text translation corpus,” in arXiv preprint arXiv:2007.10310, 2020.

[18] R. Ardila and et al., “Common voice: A massively-multilingual speech corpus,” in Proc. LREC, 2020.

[19] Xinjian Li, David R. Mortensen, Florian Metze, and Alan W Black, “Multilingual phonetic dataset for low resource speech recognition,” in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6958–6962.

[20] A. Black, “Cmu wilderness multilingual speech dataset,” in Proc. ICASSP, 2019.

[21] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. “A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings,” in Proc. ACL, 2018.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems, 2013, pp. 3111–3119.

[23] Abdul Arfat Mohammed and Venkatesh Umaashankar. “Effectiveness of hierarchical softmax in large scale classification tasks,” in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2018, pp. 1090–1094.

[24] Tuka AlHanai, Wei-Ning, and James Glass, “Development of the mit asr system for the 2016 arabic multi genre broadcast challenge..” in IEEE-SLT (Spoken Language Technologies Workshop), 2016.

[25] Seppo Enarvi, Peter Smit, Sami Virpioja, and Mikko Kurimo, “Automatic speech recognition with very large conversational finnish and estonian vocabularies,” 2017.

[26] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and et al., “Conformer: Convolution-augmented transformer for speech recognition,” arXiv preprint arXiv:2005.08100, 2020.

[27] Ashish Vaswani, Noam Shazeer, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in NIPS, 2017, pp. 5998–6008.

[28] D. Wang and X. Zhang, “THCHS-30: A free chinese speech corpus,” CoRR, vol. abs/1512.01882, 2015.

[29] R. Asiskaer, S. Yin, Z. Zhang, D. Wang, H. Askar, and F. Zheng, “THUYG-20: A free uyghur speech database,” Journal of Tsinghua University(Science and Technology), vol. 57(2): 182–187, 2017.

[30] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le, “Specaugment: A simple augmentation method for automatic speech recognition,” in INTERSPEECH, 2019.

[31] Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaooyu Chen, Lei Xie, and Xin Lei, “WeNet: Production Oriented Streaming and Non-Streaming End-to-End Speech Recognition Toolkit,” in Proc. Interspeech 2021, 2021, pp. 4054–4058.