INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Improving Multilingual Transformer Transducer Models by Reducing Language Confusions

Eric Sun[1], *Jinyu Li*[1], *Zhong Meng*[1], *Yu Wu*[2], *Jian Xue*[1], *Shujie Liu*[2], *Yifan Gong*[1]

[1]Microsoft Speech and Language Group, Redmond, WA, USA
[2]Microsoft Research Asian, Beijing, China

{sun.eric,jinyli,zhong.meng,wu.yu,jian.xue,shujliu,yifan.gong}@microsoft.com

Abstract

In end-to-end multilingual speech recognition, the hypotheses in one language could include word tokens from other languages. Language confusions happen even more frequently when language identifier (LID) is not present during inference. In this paper, we explore to reduce language confusions without using LID in model inference by creating models with multiple output heads and use sequence probability to select the correct head for output hypotheses. We propose head grouping to merge several language outputs into one head to save runtime cost. Head groups are decided by the distances among language clusters learned through language embedding vectors to separate confusable languages apart. We further propose prediction network sharing for languages from the same family. By jointly applying head grouping and prediction network sharing, training data from the same family languages is better shared while confusable languages are divided into different heads to reduce language confusions. Our experiments demonstrate that our multilingual transformer transducer models based on multi-head outputs achieve on average 7.8% and 10.9% relative word error rate reductions without LID being used in inference from one-head baseline model with affordably increased runtime cost on 10 European languages.

Index Terms: multilingual speech recognition, transformer transducer, language ID

1. Introduction

With several thousands of spoken languages in the world, the language expansion of multilingual models in Automatic Speech Recognition (ASR) has been an interesting topic that attracts lots of efforts from both industry and academia [1, 2, 3, 4, 5, 6, 7, 8, 9]. In hybrid ASR models, effective ways to build multilingual acoustic models (AMs) include hidden layer sharing [1, 2, 3] and multitask learning [4]. However, because of dedicated output layers for each individual language, hybrid multilingual models have significantly increased model size and runtime cost, making them difficult to be deployed in the industrial applications. Therefore, using multilingual models as seed models to boost the performance of low resource languages [4] has become an effective and popular way in industrial production shipping.

Recently, end-to-end (E2E) models have made rapid progress in ASR area [10, 11, 12, 13, 14, 15, 16]. E2E multilingual models have attracted lots of interests since they make ASR model training and inference even simplified. There are already lots of efforts to develop E2E multilingual models based on sequence-to-sequence models (S2S) [5, 6, 7, 8, 9], RNN transducer (RNN-T) models [17], and transformer transducer models [18]. Compared to hybrid models, E2E models are more capable of modeling the distributions of multiple languages with comparable or even better ASR performance than monolingual baseline models [5, 6, 7, 8, 17, 18, 19, 20]. Unlike a hybrid model that has one output head for each language, an E2E model only uses one output head [5, 6, 7, 8, 9, 17, 18, 19, 20] to cover all languages by passing a language identifier (LID) in form of a one-hot or learnable embedding vector to distinguish different languages.

However, in lots of applications, the ASR system is required to recognize users' speech without knowing in advance what language the user is speaking. Under such situation, LID is not available to the system. There has been little work, if any, to explore multilingual E2E models without manually feeding in LID. Instead of manually inputting LID, [19, 20] inferred LID as an embedding vector and attached it to the network input features based on RNN-T models. Since the language classifiers were only trained with acoustic features without any language text information, the inferred LID in [19, 20] was susceptible to both acoustic variations and language pronunciation similarities. Therefore, the improvement was limited. As shown in [6, 7, 8, 17, 18, 19, 20], there is a significant gap between the multilingual models trained and decoded with and without LID, because the latter system loses the LID guide to distinguish between languages. In addition, [5] proposed an attention-based S2S multilingual architecture that added language indices at the beginning of text data in training. During inference, instead of using LID as input, the system outputs both word hypotheses and LID. However, it remains unclear how the proposed idea in [5] compared with the case using LID as input since there was no related result presented in the paper.

In this paper, we propose multilingual transformer transducer models without LID input during inference in order to achieve the goal of ASR without knowing the users' language in advance. Because there is no LID guide, word token hypotheses in one language can be misrecognized as similar word tokens from other languages, which is defined as language confusions. For example, the Spanish word "sí" often appears in Italian hypotheses as a substitution error of the Italian word "sì". As we define the output of a model as its head, instead of building one-head multilingual models, our strategy is to develop multi-head models that distributes confusable languages into different heads to reduce language confusions as much as possible. In order to save runtime cost, we reduce the model heads by head grouping based on distances among learned language clusters. We further propose to have the same family languages share the same prediction network to achieve even better results. Finally, we use sequence-level log probabilities to select hypotheses from the multi-head models during decoding, which not only uses acoustic information [19, 20], but integrate language text embedding to make better output decision.

The rest of the paper is organized as follows. In Section 2, we describe the proposed methods for building high quality

http://dx.doi.org/10.21437/Interspeech.2021-1949

multilingual Transformer transducer models. The experiment results on 10 European (EU10) languages with 75 thousand hours training data are presented and analyzed in Section 3. Finally, in Section 4 we conclude and discuss our future work.

2. Multilingual Transformer Transducer

2.1. Transformer transducer model

A transducer model [11] has three components: an acoustic encoder network, a label prediction network, and a joint network as shown in Figure 1. Transducer models can use different types of models as encoders such as LSTMs in RNN-T models [11] and transformers [15, 16, 18] in transformer transducer models as shown in Figure 1. Each transformer block in the encoder network is constructed from a multi-head self-attention layer followed by a feed-forward layer. On the right side of Figure 1, the components of a transformer block are illustrated. The loss function of transducer models is the negative log posterior of output target label $\boldsymbol{y}$ given input acoustic feature $\boldsymbol{x}$ and is defined as

$$L = -\log P(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

which is calculated by the forward-backward algorithm described in [10].

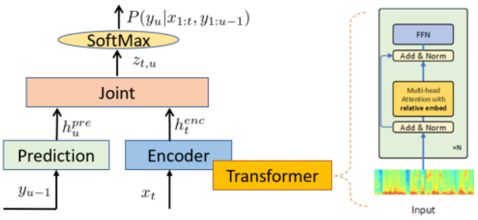

Figure 1: *Architecture of transformer transducer models*

2.2. One head vs. multi-heads

We refer to a softmax layer as the head of a multilingual model. In multilingual E2E models, multiple languages can share one head as their output layer [5, 6, 7, 8, 9, 17, 18, 19, 20] by merging the tokens from all languages into a union set as shown in Figure 2(a). Based on sentence pieces [21], language tokens among similar languages such as EU10 can be highly overlapped, which makes the model super compact as a monolingual model. However, similar tokens from different languages lead to the language confusions that make recognized hypotheses in one language include tokens from other languages. Therefore, as extra input features, LIDs are commonly used to help multilingual E2E models distinguish among different languages and improve model performance [6, 7, 8, 17, 18, 19, 20]. E2E models can also create a specific head for each language to effectively reduce the confusion between languages. As shown in Figure 2(b), language 1, 2, 3, 4 (L1, L2, L3, L4) have their own output head.

2.3. Head grouping

Even though assigning a specific head to each language can greatly reduce language confusions in multilingual models, it leads to much higher computational and memory cost since decoding needs to be performed on every head. Therefore, in order to save runtime cost with reduced the number of heads, we divide languages into several head groups, and languages in the same group share one head while they should be distinct from each other as much as possible. Therefore, the head grouping aims to gather less confusable languages together and is decided by the linear distance based on visible language embedding cluster plot with t-SNE [22]. For more detail regarding to language embedding clusters, please refer to Section 3.2. In Figure 3(a), since L1 and L2 are languages that have larger linear cluster distance, they are grouped together to share one head. Likewise, L3 and L4 share another head.

2.4. Prediction network sharing for family languages

For low and medium resource languages, one advantage of multilingual models is to share data among different languages. In E2E models, while encoders share acoustic variations, prediction networks can share tokens from different languages, especially for languages under the same language family [23] when sentence pieces are used as output units. Instead of using one prediction network for all languages, we let languages from the same language family share one prediction network. Following our head grouping strategy in Section 2.3, family languages sharing the same prediction network could be assigned to different heads according to their linear cluster distance. Note that although we encourage better data sharing for prediction network training, we still keep confusable languages apart at model output layer to reduce language confusions. For example, in Figure 3(b), since L1 and L3 are from the same language family, they share the same prediction network, but are grouped into different heads to avoid confusing output tokens with those from similar languages. The same idea applies for L2 and L4. Finally, since the number of languages could be more than the number of heads, languages from the same family could be grouped into the same head depending on their linear cluster distance.

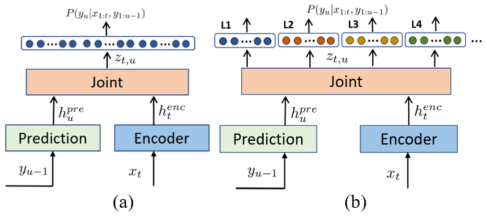

Figure 2: *(a) one-head model; (b) multi-head model*

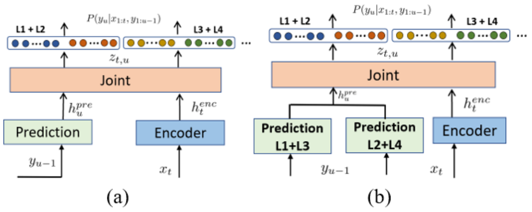

Figure 3: *(a) head grouping; (b) head grouping and prediction network sharing for languages from the same family*

2.5. Head selection for outputs

Since there are several heads in our multilingual transformer transducer models, without passing LID as the input, we need to perform decoding on every head and select the correct one as final outputs. Therefore, we select the hypothesis with the highest sequence probability as the correct one for a specific language. The sequence-level log probability is defined as

$$Score = \ P(head) * \max_{B} \log P_r(y) \tag{2}$$

where $P_r(y)$ is the probability of emitting output sequence y in beam search B, and $P(head)$ is probability of head group that is a uniform distribution. Unlike [19, 20] that the inferred LID was only trained with acoustic features, sequence-level decoding posterior probability in transducer models integrates both acoustic and language text embedding information so that it could make better decision to select correct language hypotheses as final outputs.

3. Experiments

3.1. Experiment setups

3.1.1. Languages and data

We develop our multilingual transformer transducer models for EU10 languages: German (DE), Greek (EL), English (EN), Spanish (ES), French (FR), Italian (IT), Dutch (NL), Polish (PL) Portuguese (PT), and Romanian (RO). For all these languages, Both training and test data is transcribed and anonymized with personally identifiable information removed. Test data includes both in-domain data sampled from the same distribution as training, and also out-of-domain data that is different from training. The training and test data amount per language is summarized in Table 1.

Table 1: *Train and test data per language (in hours)*

Language	Train	Test
DE	4,894	38.1
EL	2,191	20.2
EN	38,592	207.6
ES	7,587	33.1
FR	5,957	33.3
IT	6,581	19.2
NL	883	6.1
PL	2,115	5.1
PT	4,304	16.6
RO	1,902	13.0
Total	75,007	392.3

3.1.2. Model structure and training configuration

In our transformer transducer models, 18 transformer blocks with 320 hidden nodes, 8 attention heads, and 2048 feedforward nodes are used as the encoder; 2 LSTM layers with 1024-dimensional embedding and hidden layer are used in the prediction network. All our experiments use 80-dimensional log-Mel filterbanks with 25 millisecond (ms) windows and 10ms shift. Two convolutional layers are applied to get features with 40ms sampling rate. The input acoustic feature sequence is segmented into chunks with a chunk size of 18 in our experiments and chunks are not overlapped. Therefore, the maximum lookahead is 720ms. In addition, we also apply 4 left chunks to leverage history acoustic information in our training. An effective mask strategy to truncate history and allow limited future lookahead information has been designed as in [24]. All the models are trained from scratch and with mixed precision for efficient training. The learning rate warmup strategy is the same as in [25]. Each training mini-batch consists of utterances from all languages, sampled according to their natural training data distribution. Around 10k sentence pieces [21] from all speech transcribed text data of EU10 are used as token units in all experiments.

3.2. Results

3.2.1. Baseline multilingual models with and without LID

We train a multilingual model as the baseline for our experiments by just mixing all training data from EU10 without passing LID. We then train another multilingual model with the same model architecture, but append a 10-dimensional oracle LID one-hot vector with acoustic features as input. Table 2 shows the multilingual model trained and decoded with LID is 19.2% significantly better than the model without LID being used in training and decoding, especially on the languages that have mismatch between training and test data such as FR, IT, NL and PL. In Table 3, we show examples of language confusion errors from Italian to Spanish (marked with red color) for the model without LID. These confusion errors are able to be fixed by the LID model in Table 3.

3.2.2. Head grouping

We separate confusable languages into different groups to avoid language confusions. The visible language clusters based on learned embeddings from the prediction network of baseline multilingual model without LID are demonstrated using t-SNE [22] in Figure 4(a). We maximize the linear distance among language clusters and divide them into two groups as 1) DE, NL, IT, PT and EL (green shape); and 2) EN, PL, ES, FR, and RO (blue shape) as shown in Figure 4(b), and three groups as 1) EN, ES, and EL (blue shape); 2) PT, NL, and FR (green shape); and 3) IT, PL, DE, and RO (brown shape) as shown in Figure 4(c). In Table 2, word error rates (WERs) of multilingual models are shown with two heads (2H) for two language groups as in Figure 4(b), three heads (3H) for three language groups as in Figure 4(c), and ten heads (10H). All these 2H, 3H and 10H models are decoded with provided oracle LID. 10H model is 1.9% relatively better while 2H and 3H models are 16.0% and 8.3% relatively worse than the baseline LID model. 10H model is most effective to reduce language confusions. As discussed in Section 2.5, without passing LID as input during inference, we also evaluated the multi-head models with sequence-level log probabilities to select correct head for output hypotheses. In Table 2, ten head model with probability selection (10H prob.), two head model with probability selection (2H prob.), and three head model with probability selection (3H prob.) are 11.4%, 7.8%, and 3.1% better than the baseline model without LID, respectively.

Table 2: *WERs for EU10 based on different multilingual models*

Model	DE	EL	EN	ES	FR	IT	NL	PL	PT	RO	Avg
Baseline without LID	18.2	17.8	10.7	19.8	27.0	21.6	24.4	24.0	14.1	15.6	19.3
Baseline with LID	16.2	17.5	10.5	16.1	17.4	15.3	17.7	17.7	13.0	14.6	15.6
2H	17.2	17.4	10.4	17.1	23.6	19.2	23.0	25.0	13.8	15.0	18.1
3H	16.3	17.2	10.3	16.7	23.4	14.8	21.7	20.6	13.3	14.8	16.9
10H	16.0	17.1	10.2	15.6	19.0	13.7	17.7	17.4	12.7	13.7	15.3
2H prob.	17.7	17.5	10.6	**18.0**	24.2	20.0	23.6	26.1	13.4	15.1	18.7
3H prob.	**17.0**	**17.4**	**10.6**	18.2	23.9	**16.3**	23.2	22.4	**14.0**	14.8	17.8
10H prob.	18.0	18.1	11.4	18.4	**21.4**	16.5	**19.4**	**19.5**	14.3	**13.8**	17.1
3P+3H	16.1	16.4	10.0	15.8	22.0	13.9	20.3	19.7	12.9	14.2	16.1
3P+3H prob.	16.9	16.5	10.4	17.1	22.6	15.6	21.9	21.8	13.7	14.3	17.1

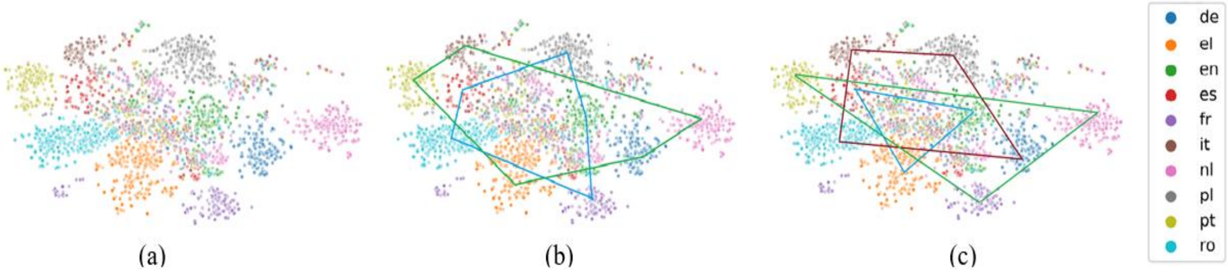

Figure 4: *(a) language clusters; (b) two head groups; (c) three head groups*

3H prob. model is clearly better than 2H prob. model. Although 10H prob. model gets better averaged WER than 3H prob. model, on several individual languages such as DE, EL, EN, IT and PT, 3H prob. model obtains better WERs. In addition, 10H prob. model has much higher computational and memory cost in runtime. Therefore, 3H prob. model is the best choice with a good balance between runtime cost and model performance. By the way, we also jointly trained one-head multilingual model and a LID classifier with multi-task learning to infer LID during decoding as in [20], but got much worse results than our current method.

Table 3: *Language confusion examples for Italian*

Ref (Italian)	With LID	Without LID
prego sì	prego sì	prego sí
menù della cena	menù della cena	menú della cena
otto e ventiquattro	otto e ventiquattro	otto y veinticuatro
sedici e venticinque	sedici e venticinque	se dice venticinque

3.2.3. *Prediction network sharing for family languages*

We further improve 3H model with three shared prediction networks. Languages share the same prediction network if they are in the same language family [23]. The three groups of languages that share the same prediction network are 1) EN, DE, NL, and FR; 2) PT, IT, ES, RO, and PL; and 3) EL. Actually, FR should have been in the second group according to language family [23]. However, from Figure 4(a), FR cluster is closer to the languages in the first group. In addition, EL has different characters from other 9 languages, and therefore form a group by itself. We integrate the prediction network sharing with head grouping in Section 3.2.2 to construct a model with three prediction networks and three heads (3P+3H). All the three prediction networks have the same model structure as described in Section 3.1.2. In addition, languages that share the same prediction network could also be grouped into the same head such as NL and FR since their clusters are far away from each other as shown in Figure 4(c). In Table 2, 3P+3H model further improves 3H model by 4.7% relatively, and is only 3.2% relatively worse than baseline LID model. 3P+3H model with probability selection (3P+3H prob.) improves 3.9% relatively over 3H prob. model, and is 10.9% relatively better than baseline model without LID.

3.2.4. *Runtime cost discussion and head selection accuracy*

10H prob. model has higher runtime computational cost and language confusions than 3H prob. model since it has to do ten decoding while 3H prob. model only runs three decoding in parallel. In Table 4, 10H prob. model obtains 90.2% accuracy that is absolute 2.6% worse than 3H prob. model, which explains 10H model is 9.5% better than 3H model, while 10H prob. model is only 3.9% better than 3H prob. model. In addition, 3P+3H prob. model needs to run 9 decoding in parallel, but it obtains 92% accuracy that is much better than 10H prob. model, which means better data sharing for prediction networks can help reduce language confusions. In terms of memory footprint, 3H model, 3P+3H model, and 10H models have 112.5, 132.5, and 212.5 million float parameters, respectively. Overall, 3P+3H prob. model has less runtime cost than 10H prob. model but with same averaged WER and better scalability for multilingual language extension.

Table 4: *Accuracies of head selection for outputs*

Language	10H prob.	3H prob.	3P+3H prob.
DE	86.9	92.3	91.2
EL	92.7	97.3	97.4
EN	88.8	92.9	92.9
ES	85.5	88.5	88.7
FR	88.0	92.9	91.7
IT	88.3	92.3	91.2
NL	91.4	87.9	86.7
PL	94.5	94.0	92.6
PT	87.6	91.1	90.2
RO	98.2	98.8	97.8
Avg	90.2	92.8	92.0

4. Conclusions and Future Work

In this paper, we explored multi-head transformer transducer multilingual models to reduce language confusions without using LID in model training. We proposed to: 1) apply head grouping to separate confusable languages into different heads; 2) apply family language prediction network sharing to encourage better data sharing for languages in the same family; 3) apply sequence-level probability to automatically select the right head for output hypotheses. Our best models achieved 7.8% and 10.9% relative WER reductions from baseline model without LID on EU10. In the future, we plan to further improve head selection accuracies for 3H and 3P+3H prob. models in order to achieve even better model performance.

5. References

[1] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In ICASSP, pages 7304–7308, 2013.

[2] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in ICASSP, 2013.

[3] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in ICASSP, 2013.

[4] D. Chen and B. Mak, “Multitask learning of deep neural networks for low-resource speech recognition,” in IEEE/ACM Transactions.

[5] Watanabe, S., Hori, T., and Hershey, J. R. “Language independent end-to-end architecture for joint language identification and speech recognition,” in Proc. ASRU, pp. 265-271, 2017.

[6] S. Toshniwal, T. Sainath, R. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, “Multilingual speech recognition with a single end-to-end model,” in ICASSP, 2018.

[7] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, “Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes,” in ICASSP, 2018.

[8] V. Pratap, A. Sriram, P. Tomasello, A. Hannun, V. Liptchinsky, G. Synnaeve, and R. Collobert, “Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters,” arXiv:2007.03001, 2020.

[9] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, “Massively multilingual adversarial speech recognition,” arXiv:1904.02210, 2019.

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

[11] Alex Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1211.3711, 2012.

[12] He, Y., Sainath, T.N., Prabhavalkar, R., et al. “Streaming end-to-end speech recognition for mobile devices,” in *Proc. ICASSP*, 2019.

[13] J. Li, R. Zhao, H. Hu, and Y. Gong, “Improving RNN transducer modeling for end-to-end speech recognition,” in Proc. ASRU, 2019.

[14] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, “On the comparison of popular end-to-end models for large scale speech recognition,” in Proc. Interspeech, 2020.

[15] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech

[16] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, “Transformer-transducer: End-to-end speech recognition with self-attention,” 2019.

[17] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, “Large-scale multilingual speech recognition with a streaming end-to-end model,” Interspeech 2019, Sep 2019.

[18] Y. Zhu, P. Haghani, A. Tripathi, B. Ramabhadran, B. Farris, H. Xu, H. Lu, H. Sak, I. Leal, N. Gaur, P. J. Moreno, Q. Zhang, “Multilingual speech recognition with self-attention structured Parameterization,” in Interspeech, 2020.

[19] A. Waters, N. Gaur, P. Haghani, P. Moreno, Z. Qu, “Leveraging language ID in multilingual end-to-end speech recognition,” ASRU, 2019

[20] S. Punjabi, H. Arsikere, Z. Raeesy, C. Chandak, N. Bhave, A. Bansal, M. Muller, S. Murillo, A. Rastrow, S. Garimella, ¨ et al., “Streaming end-to-end bilingual asr systems with joint language identification,” arXiv preprint arXiv:2007.03900, 2020.

[21] https://github.com/google/sentencepiece

[22] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.

[23] https://en.wikipedia.org/wiki/Languages_of_the_European_Union

[24] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, “Developing real-time streaming transformer transducer for speech recognition on large-scale dataset,” arXiv preprint arXiv:2010.11395, 2020.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, pages 5998–6008, 2017.