INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Distant Supervision for Polyphone Disambiguation in Mandarin Chinese

Jiawen Zhang[1,2], *Yuanyuan Zhao*[3], *Jiaqi Zhu*[1,2,4], *Jinba Xiao*[3]

[1]Institute of Software, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
[3]Kwai, Beijing, China
[4]Zhejiang Lab, China

jiawen2019@iscas.ac.cn, zhaoyuanyuan@kuaishou.com, zhujq@ios.ac.cn

Abstract

Grapheme-to-phoneme (G2P) conversion plays an important role in building a Mandarin Chinese text-to-speech (TTS) system, where the polyphone disambiguation is an indispensable task. However, most of the previous polyphone disambiguation models are trained on manually annotated datasets, which are suffering from data scarcity, narrow coverage, and unbalanced data distribution. In this paper, we propose a framework that can predict the pronunciations of Chinese characters, and the core model is trained in a distantly supervised way. Specifically, we utilize the alignment procedure used for acoustic models to produce abundant character-phoneme sequence pairs , which are employed to train a Seq2Seq model with attention mechanism. We also make use of a language model that is trained on phoneme sequences to alleviate the impact of noises in the auto-generated dataset. Experimental results demonstrate that even without additional syntactic features and pre-trained embeddings, our approach achieves competitive prediction results, and especially improves the predictive accuracy for unbalanced polyphonic characters. In addition, compared with the manually annotated training datasets, the auto-generated one is more diversified and makes the results more consistent with the pronunciation habits of most people.

Index Terms: Polyphone disambiguation, Grapheme-to-phoneme conversion, Text-to-Speech, Distant supervision

1. Introduction

The text-to-speech (TTS) system is an essential component in the human-computer voice interaction framework, which aims to "translate" natural language texts into speech. Although most of Chinese characters have a fixed pronunciation, there are some special cases, i.e. the polyphonic characters. They have different pronunciations in different contexts, and the choice of pronunciations would have a huge impact on the semantics. Therefore, identifying a correct pronunciation for a polyphonic character according to its context, called polyphone disambiguation, is an important task for the Chinese TTS system.

There are many approaches proposed to address the polyphone disambiguation problem. They can be divided into two categories: rule-based approaches and data-driven approaches. The former approaches heavily rely on human effort to work out pronunciation dictionaries and rules. For the latter, some researches proposed to learn statistical rules from data instead of using hand-crafted ones [1, 2], and some other works treated polyphone disambiguation as a classification task based on syntactic features [3, 4, 5]. More recently, due to the outstanding performance, neural networks have been employed to extract contextual features in an increasing number of works [6, 7, 8]. However, data-driven approaches face the following challenges: (1) the lack of publicly available datasets. Few public datasets are targeting at polyphone disambiguation, and the size of them are usually small since the manual labeling is costly. (2) The unbalanced data distribution. The data imbalance problem naturally exists in Chinese, as the statistical analysis of real data suggested that the curve of Chinese characters' occurrence frequency is in a long-tailed shape [9]. Besides, it can also be observed that many polyphonic characters follow a regularity that one pronunciation occupies the overwhelming majority in their overall occurrences. (3) The biases among different annotators. Due to the diverse usage of polyphonic characters, different speaking habits of annotators are likely to lead to biases in the annotated training data [10].

Distant supervision is a paradigm widely used for the relation extraction task [11], which annotates the unstructured texts automatically with relational labels on the basis of a knowledge base. Following this idea, we propose a framework for phoneme prediction, and especially suitable for polyphone disambiguation. The input is a Chinese character sequence and the output is the corresponding phoneme sequence. The core model is trained in a distantly supervised way.

This framework is composed of three modules: the character-phoneme transformation module, the distantly supervised data generation module, and the reranking module. The character-phoneme transformation module adopts the sequence-to-sequence (Seq2Seq) encoder-decoder model to capture the context of input characters as well as the correlation between input texts and output phonemes. The training data are generated with the aid of the distantly supervised data generation module, which adopts directly the alignment procedure used for the acoustic model, and the acoustic features in this module enable Chinese characters to align with their phonemes automatically. In this way, our approach can obtain numerous training data without labeling effort, and also avoid the subjectivity of manual annotations. Nevertheless, the distant supervision suffers from an obvious problem: the noise. To alleviate its impact, inspired by [12], we introduce the reranking module. It employs a language model trained on phoneme sequences to re-score and re-order the n-best candidate phoneme sequences generated by the character-phoneme transformation module.

We conduct experiments on both auto-generated and manually annotated test datasets. The results indicate that: (1) although without syntactic features and pre-trained embeddings, our approach presents competitive performances; (2) when dealing with some long-tailed characters, our approach achieves better performance than the same model trained solely on manually annotated data; (3) the phonemes are distantly annotated by the speech data collected in real scenarios, which makes the results closer to the pronunciation habits of most people.

http://dx.doi.org/10.21437/Interspeech.2020-2427

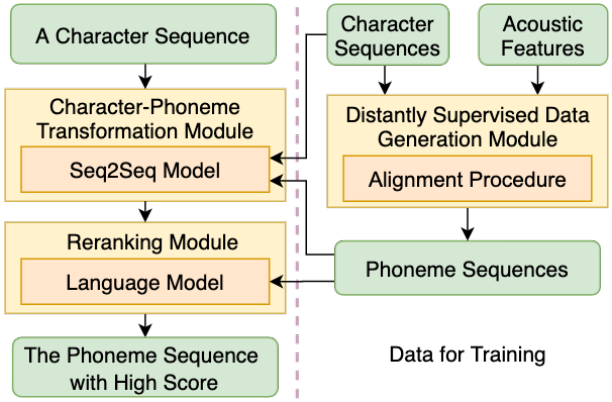

Figure 1: *Proposed phoneme prediction framework.*

2. Chinese Polyphonic Characters

The Pinyin is commonly used in Mandarin Chinese to indicate a character's pronunciation, and it usually consists of several phonemes with a tone. For example, the pronunciation of the character "你" can be represented as "ni3", which consists of phonemes "n" and "i" with a tone "3". For most of Chinese characters, the pronunciation is fixed, i.e. there is only a single pinyin corresponding to a character. However, for some special cases, the pronunciation is manifold, and the selection of pronunciations would affect the meaning of the whole sentence. As shown in Table 1, in most situations, the pronunciation of a polyphonic character can be determined based on its neighbors and semantic role in the sentence.

Table 1: *Examples of polyphonic characters and their English translation in different contexts.*

Character sequence	English translation
我不喜欢抽雪**茄 (jia1)**但是我喜欢吃番*茄 (qie2)*	I don't like to smoke **cigars**, but I like to eat *tomatoes*
他们两人之**间 (jian1)**的友谊从来没有*间 (jian4)*断过	The friendship **between** them never *interrupted*

3. Method

3.1. Framework overview

The proposed framework consists of three modules: the distantly supervised data generation module, the character-phoneme transformation module, and the reranking module. As shown in Figure 1, the distantly supervised data generation module is fed with Chinese character sequences and their corresponding acoustic features, and then outputs the phoneme sequences automatically. Afterwards, those character-phoneme sequence pairs are used to train the character-phoneme transformation module. To alleviate the impact of noises, a language model is employed to re-score and re-rank the prediction results. The predicted phoneme sequence with the highest score would be the final results. In the following subsections, we will present each module of the framework in detail.

3.2. Distantly supervised data generation module

The distantly supervised data generation module is introduced to obtain the phoneme sequences corresponding to the given character sequences automatically. To achieve this, we employ the speech-text force alignment procedure used for acoustic models, which takes acoustic features and character sequences as input, and the phoneme sequence aligned to the character sequence as output. Since it is a fully functional and well-performing module, we adopt it directly, and a minimal Gated Recurrent Unit with Input Projection layer (mGRUIP) [13] is harnessed to model the future temporal contexts efficiently. Then, the character-phoneme sequence pairs generated by this module are adopted as the training data of the following models.

3.3. Character-phoneme transformation module

Most of the previous work regarded polyphone disambiguation as a classification problem for single characters that utilized neural networks to extract the contextual features of character sequences. However in this way, the information available in the phoneme sequence was never exploited. It is noteworthy that the Grapheme-to-phoneme (G2P) conversion is similar to a translation task. Apart from learning parameters from the source language and data pairs, the target language also contains many learnable patterns. Therefore, we apply the Seq2Seq model, which is trained upon the whole character sequences and phoneme sequences, rather than merely the features of the source character sequences.

3.3.1. Data preprocessing

Before training the Seq2Seq model, the input elements (i.e. characters and phonemes) should be embedded into a distributional space. Following the settings in [14], a dictionary that contains the mappings of characters and phonemes to their vector representations is built based on our auto-generated dataset.

Note that the usage of pre-trained embeddings (e.g., Word2Vec [15]) or linguistic features to encode texts' representations would probably lead to better performances, but we only apply the simplest token embeddings here to examine the prediction results without any additional prior knowledge.

3.3.2. The Seq2Seq model

We use a Seq2Seq model to predict the corresponding phoneme sequence of the input character sequence. More specifically, given a character sequence x, for each possible phoneme sequence y, the model computes the probability $p(y|x)$.

For the implementation of the Seq2Seq model, we adopt the CNN architecture proposed by Gehring et al. [16]. It is equipped with position embeddings, convolutional block structure, as well as gated linear units, and a separate attention mechanism for each decoder layer is introduced. Compared with recurrent networks, the convolutional approach has the advantage of discovering compositional structures in the sequences.

As alternatives, we also harness the Long Short-Term Memory (LSTM) network [17] with the global attention mechanism [18], which is capable of solving the explosion and vanishing gradient problems when handling sequential data, as well as the Transformer network [19], which replaces the LSTM structure with solely attention mechanism and has achieved impressive performances.

3.4. Reranking module

Since our training data is automatically generated, it is likely to contain many noises. Besides, in many cases, the paired data are much fewer than the unpaired data, e.g., the paired Chinese-English translations are rare, but the English texts are abundant. To this end, inspired by the neural noisy channel model [12], we train a Transformer-based language model $p(y)$ to re-score the n-best candidate phoneme sequences generated by the Seq2Seq model, which aims at making full use of the unpaired data to alleviate the noise problem. Then, the predicted scores of the Seq2Seq model and the language model are linearly combined as:

$$\log p(y|x) + \lambda \log p(y) \quad (1)$$

where λ is a fixed weight, and will be made tunable in the future work.

4. Experiment

4.1. Dataset

We use three datasets to test the performance of our approach. The first one is auto-generated by the data generation module, and the rest are human-annotated. The statistics of them are shown in the Table 2.

AUTO dataset The auto-generated (AUTO) dataset, which includes 7,082,058 sentences, is annotated by the data generation module, and 79.23% of those sentences contain polyphonic characters, which is closer to the real scenarios. We randomly select 10,000 data samples for the validation set and the test set respectively, and the rest are used for training.

MANU test set We also conduct the evaluation on a manually annotated (MANU) test set, which includes 146, 338 sentences, and the phoneme of each polyphonic character is annotated by native Mandarin Chinese speakers.

CPP dataset Lately, Park et al. published a bidirectional LSTM-based conversion package g2pM[1] that was trained upon a human-annotated benchmark dataset, named Chinese Polyphones with Pinyin (CPP). It includes 99, 264 sentences and is open to the public. In this dataset, only one polyphonic character is annotated in each sentence, even if two or more polyphonic characters occur. Thus, to make this dataset applicable to our framework, besides labeled Chinese characters, we also leverage g2pM to annotate all the unlabeled Chinese characters.

Table 2: *Statistics of the datasets used in our experiment.*

	AUTO	MANU	CPP
Total	7, 082, 058	146, 338	99, 264
Training	7, 062, 058	-	79, 117
Validation	10, 000	-	9, 893
Test	10, 000	146, 338	10, 254

4.2. Experimental settings

Baselines For comparison, we take a commonly used dictionary-based conversion package Pypinyin[2] and the conversion package g2pM mentioned above as two baselines.

Our approach For the Seq2Seq model, we test the performances of three networks (LSTM, Transformer, and CNN). To verify the effects of the language model, we evaluate the prediction module itself and the results re-ranked by the language model (+LM). In addition, we also trained the prediction module based on the CPP training set, marked with (CPP), for a fair comparison with g2pM.

- **Distantly supervised data generation module** It is trained on around 8000h internal Mandarin speech data with LF-MMI objective function, which is computed at a frequency of 33Hz. It contains 5 layers, each hidden layer consists of 1024 units and the input projection layer has 512 units. The input feature at time step t is a splicing from frame $t-2$ to frame $t+2$, and the target delay is 50ms. The output is context-dependent phonemes of size 2408. After testing, the character error rate (CER) of the G2P model with pre-trained 4-gram LM achieves 2.91 upon the Aishell test corpus [20], which indicates that the alignment procedure used here is capable of producing the phoneme sequences with high accuracy.
- **Character-phoneme transformation module** We adopt the FAIRSEQ sequence modeling toolkit [14] developed by Facebook, which provides implementations of various Seq2Seq models within the encoder-decoder framework. We set the number of hidden units to 512 and the batch size to 500. The number of layers is 2, 6, and 20 for LSTM, Transformer, and CNN respectively, where the encoder and decoder share the same configuration. We set the learning rate to 0.0005, and employ the cross-entropy loss with NAG optimizer [21] for CNN architecture, and the Adam optimizer [22] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ for the LSTM-based and Transformer-based models. The width of the beam search is 5, and the 50-best predictions of character-phoneme transformation module are fed into the reranking module.
- **Reranking module** The language model in the reranking module is trained on the AUTO dataset based on the implementation in [23], and shares the same parameter configuration with the Transformer in the previous module. After validation, we set λ in Equation 1 to 1.0.

Table 3: *Accuracy on three test datasets for different models.*

Model	AUTO	MANU	CPP
Pypinyin	89.25 %	82.53 %	86.64 %
g2pM	87.33 %	82.93 %	94.90 %
LSTM	91.71 %	81.31 %	77.83 %
Trans.	96.56 %	84.64 %	83.18 %
CNN	96.86 %	86.01 %	89.52 %
LSTM + LM	95.81 %	81.28 %	79.07 %
Trans. + LM	96.61 %	84.66 %	82.90 %
CNN + LM	**96.88 %**	**86.20 %**	89.49 %
CNN (CPP)	87.46 %	79.25 %	97.20 %
CNN+LM (CPP)	87.76 %	82.93 %	**97.51** %

4.3. Results and analysis

4.3.1. Overall evaluation results

Table 3 compares the accuracy of different models upon different test datasets. We can observe that:

(1) Although the AUTO training set contains many noises, we find that our approach has a better overall performance than

[1] https://github.com/kakaobrain/g2pM

[2] https://github.com/mozillazg/python-pinyin

Table 4: *Comparison of our approach trained with and without the AUTO dataset. Both models are tested upon the MANU test set.*

Polyphonic character	Statistics (phoneme: frequency)						Accuracy	
	CPP			CPP+AUTO			Trained on CPP	Trained on CPP+AUTO
糊	hu2: 33	hu1: 0		hu2: 2221	hu1: 430		32.82 %	**61.83 %**
帖	tie1: 2	tie3: 151	tie4: 0	tie1: 536	tie3: 2508	tie4: 16	62.96 %	**66.67 %**
踏	ta4: 158	ta1: 2		ta4: 1706	ta1: 790		77.50 %	**97.50 %**
重	chong2: 58	zhong4: 99		chong2: 18349	zhong4: 47635		54.68%	**82.31 %**
应	yin4: 129	yin1: 31		yin4: 26380	yin1: 41543		39.89 %	**87.80 %**
少	shao4: 54	shao3: 106		shao4: 9260	shao3: 70251		**80.28 %**	76.44 %
得	de2: 138	dei3: 0		de2: 70497	dei3: 25321		52.35 %	**81.60 %**
Overall	79, 117			7, 141, 175			82.93 %	**85.13 %**

baselines. That indicates the damage caused by noise is not enough to counteract the benefits from its volume and coverage. In addition, the CNN-based model outperforms the LSTM-based and Transformer-based models. A possible reason is that the LSTM structure cannot capture the reversed features, and compared with the Transformer network's attention mechanism, the sliding window in CNN is good at capturing local structures.

(2) Overall, the language model indeed improves the accuracy and the degree of improvement varies for different networks. For networks with poorer predictive accuracy, the usage of language models leads to greater improvements. That shows the language model can make up for the deficiency in the prediction model, and we believe that if more accurate phoneme sequences are available, its effect will be more significant.

(3) Both trained on the CPP training set, the CNN+LM model performs better than the g2pM model upon the CPP test set. Therefore, when the annotated training data is accessible, integrating auto-generated data can achieve better results.

(4) The accuracy upon the MANU test dataset is lower than that in the AUTO dataset. It is not surprising since the AUTO training set and the AUTO test set are homogeneous, while the manually annotated test sets contain less noise.

4.3.2. Training data analysis

In this section, we give some examples to show how our approach improves the performances of long-tail polyphonic characters and those without strictly acknowledged pronunciation patterns. We train our character-phoneme transformation module (with CNN network) and reranking module on two datasets: one is the CPP dataset, denoted as **CPP**, and the other is the combination of the CPP and the AUTO training sets, denoted as **CPP+AUTO**. Both models are tested upon the MANU test set. Table 4 shows the statistics and the experimental results for several representative polyphonic characters.

For the long-tail polyphonic characters (e.g., "糊") and the characters with serious data imbalance (e.g., "帖" and "踏"), the introduction of AUTO training data greatly improves the predictive accuracy. This can be explained that the AUTO dataset has a large base, which provides more samples for the minority of pronunciations to augment the data. Moreover, even for the polyphonic characters with modest data imbalance problems (e.g., "重" and "应"), the added AUTO training set can promote the accuracy as well, since additional data samples enable the model to learn various patterns of the character. However, we also observe that when a character has a balanced distribution and its size is large enough for training (e.g., "少"), adding a large number of unbalanced data would impair the accuracy. After further checking, we find that such cases only take up a small proportion, and the loss of the accuracy is small, so it is acceptable in view of the profit brought by the AUTO training set.

When there is no significant pronunciation difference in the usage of the polyphonic characters in daily life (e.g., "得"), the labeled pronunciations in human-annotated datasets depend entirely on the annotators' speaking habits. We observed that the annotators of the CPP dataset have labeled all the pronunciations of "得" as "de2", whereas under many situations, "dei3" is the more authentic answer. In the AUTO dataset, the ground truth is obtained via the real speech data, while less related to the speaking habit of a specific annotator. In other words, it is closer to the way most people speak.

5. Conclusion

In this paper, we propose a framework trained in a distantly supervised way for Mandarin Chinese polyphone disambiguation. Instead of manual annotation, given some character sequences, our framework takes acoustic features as the basis to automatically generate the corresponding phoneme sequences. To take full advantage of the auto-generated phoneme sequences, we train the character-phoneme transformation module in the Seq2Seq manner and leverage a language model to alleviate the impact of noises. Experimental results show that our approach is competitive to the dictionary-based baseline and the models trained on human-annotated data. Especially, the integration of abundant auto-generated training data improves the predictive accuracy of some long-tailed and ambiguous polyphonic characters. In addition, our results are more consistent with the pronunciation habits of most people, but not specific annotators, which is essential for a Mandarin Chinese TTS system to produce natural speech.

6. Acknowledgements

This work was supported by the National Key R&D Program of China (2018YFC0116703).

7. References

[1] H. Zhang, J. Yu, W. Zhan, and S. Yu, "Disambiguation of Chinese polyphonic characters," in *The First International Workshop on MultiMedia Annotation (MMA2001)*, vol. 1. Citeseer, 2001, pp. 30–31.

[2] Z.-R. Zhang, M. Chu, and E. Chang, "An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese," in *International Symposium on Chinese Spoken Language Processing*, 2002, pp. 59–62.

[3] F. Liu and Y. Zhou, "Polyphone disambiguation based on tree-guided TBL," *Computer Engineering and Applications*, vol. 47, no. 12, pp. 137–140, 2011.

[4] F. Z. Liu and Y. Zhou, "Polyphone disambiguation based on maximum entropy model in Mandarin grapheme-to-phoneme conversion," in *Key Engineering Materials*, vol. 480. Trans Tech Publ, 2011, pp. 1043–1048.

[5] D. Dai, Z. Wu, S. Kang, X. Wu, J. Jia, D. Su, D. Yu, and H. Meng, "Disambiguation of Chinese polyphones in an end-to-end framework with semantic features extracted by pre-trained BERT," in *Annual Conference of the International Speech Communication Association*, 2019, pp. 2090–2094.

[6] B. Yang, J. Zhong, and S. Liu, "Pre-trained text representations for improving front-end text processing in Mandarin text-to-speech synthesis," in *Annual Conference of the International Speech Communication Association*, 2019, pp. 4480–4484.

[7] C. Shan, L. Xie, and K. Yao, "A bi-directional LSTM approach for polyphone disambiguation in Mandarin chinese," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2016, pp. 1–5.

[8] Z. Cai, Y. Yang, C. Zhang, X. Qin, and M. Li, "Polyphone disambiguation for Mandarin Chinese using conditional neural network with multi-level embedding features," in *Annual Conference of the International Speech Communication Association*, 2019, pp. 2110–2114.

[9] L. Q. Xie, H. P. Fang, and Y. S. Jin, "Statistical analysis for standard Chinese common-words, syllables and phoneme system," *Journal of Northwest University for Nationalities*, vol. 51, no. 11, pp. 35–39, 2012.

[10] A. Søgaard, B. Plank, and D. Hovy, "Selection bias, label bias, and bias in ground truth," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts*, 2014, pp. 11–13.

[11] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 1003–1011.

[12] L. Yu, P. Blunsom, C. Dyer, E. Grefenstette, and T. Kocisky, "The neural noisy channel," in *International Conference on Learning Representations*, 2017, pp. 1–13.

[13] J. Li, X. Wang, Y. Zhao, and Y. Li, "Gated recurrent unit based acoustic modeling with future context," in *Annual Conference of the International Speech Communication Association*, 2018, pp. 1788–1792.

[14] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "FAIRSEQ: A fast, extensible toolkit for sequence modeling," in *North American Chapter of the Association for Computational Linguistics*, 2019, pp. 48–53.

[15] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996.

[16] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*, 2017, pp. 1243–1252.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[19] 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*, 2017, pp. 5998–6008.

[20] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[21] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence O(1/k^2)," in *Doklady ANSSSR*, vol. 269, 1983, pp. 543–547.

[22] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *International Conference for Learning Representations*, 2014, pp. 1–15.

[23] K. Yee, N. Ng, Y. N. Dauphin, and M. Auli, "Simple and effective noisy channel modeling for neural machine translation," in *Empirical Methods in Natural Language Processing*, 2019, pp. 5695–5700.