Samsung R&D Institute Philippines at WMT 2023

Jan Christian Blaise Cruz
Samsung R&D Institute Philippines
jcb.cruz@samsung.com

Abstract

In this paper, we describe the constrained MT systems submitted by Samsung R&D Institute Philippines to the WMT 2023 General Translation Task for two directions: en→he and he→en. Our systems comprise of Transformer-based sequence-to-sequence models that are trained with a mix of best practices: comprehensive data preprocessing pipelines, synthetic backtranslated data, and the use of noisy channel reranking during online decoding. Our models perform comparably to, and sometimes outperform, strong baseline unconstrained systems such as **mBART50 M2M** and **NLLB 200 MoE** despite having significantly fewer parameters on two public benchmarks: FLORES-200 and NTREX-128.

1 Introduction

This paper describes Samsung R&D Institute Philippines's submission to the WMT 2023 General Translation task. We participate in two translation directions: en→he and he→en, submitting two **constrained** single-direction models based on the Transformer (Vaswani et al., 2017) sequence-to-sequence architecture. We employ a number of best practices, using a comprehensive data preprocessing pipeline to ensure parallel data quality, create synthetic data through carefully-curated backtranslation, and use reranking methods to select the best candidate translations.

Our systems achieve strong performance on public benchmarks: 44.24 BLEU and 33.77 BLEU for FLORES-200 and NTREX-128 en→he, respectively, and; 42.42 BLEU and 36.89 BLEU on FLORES-200 and NTREX-128 he→en, respectively. Our systems outperform **mBART50 M2M** and slightly underperform against **NLLB 200 MoE** despite having significantly less parameters compared to these unconstrained baselines.

We detail our data preprocessing, model training, data augmentation, and translation methodology. Additionally, we illustrate hyperparameter sweeping setups and study the effects of hyperparameters during online decoding with reranking.

2 Methodology

2.1 Data Preprocessing

Given that a significant portion of the training dataset is synthetically-aligned, we need to use a comprehensive data preprocessing pipeline to ensure good translation quality. In particular, we use a combination of heuristic-based, ratio-based, and embedding-based methods to filter our data.

Heuristic-based The following heuristic-based filters based on Cruz and Cheng (2021) are used before applying the others:

- **Language Filter** – We use use `pycld3`[1] to filter out sentence pairs where one or both sentences have more than 30% tokens that are neither English nor Hebrew.
- **Named Entity Filter** – We use NER models (Bareket and Tsarfaty, 2021; Yang and Zhang, 2018) to check if both sentences in a pair have matching entities (if any). Pairs that contain entities that do not match are removed.
- **Numerical Filter** – If one sentence in a pair has a number (ordinal, date, etc.), we also check the other sentence if a matching number is present. If a match is not detected, the pair is removed.

Ratio-based We employ ratio-based filters on tokenized sentence pairs following Cruz and Sutawika (2022) and Sutawika and Cruz (2021). We first tokenize using SacreMoses[2] then apply the following ratio-based filters:

[1] https://pypi.org/project/pycld2/
[2] https://github.com/alvations/sacremoses

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 103–109
December 6–7, 2023. ©2023 Association for Computational Linguistics

	Pairs	Words (en)	Words (he)
Original	72,459,348	701,991,594	566,555,530
Original Filtered	48,278,395	385,975,984	312,639,617
Synthetic en→he	10,000,000	165,595,289	145,849,940
Synthetic en→he Filtered	7,143,725	115,239,312	95,954,020
Synthetic he→en	73,278,018	1,471,827,973	1,056,677,671
Synthetic he→en Filtered	47,372,416	659,409,236	541,376,459

Table 1: Corpus Statistics. "Filtered" refers to the number of pairs / words that remain after the filtering script is applied to the dataset. Note that "Words" is an approximation gathered by using the `wc -l *` command on the plaintext files.

- **Length Filter** – We remove pairs containing sentences with more than 140 characters.
- **Token Length Filter** – We remove pairs that contain sentences with tokens that are more than 40 characters long.
- **Character to Token Ratio** – We remove pairs where the ratio between character count and token count in at least one sentence is greater than 12.
- **Pair Token Ratio** – We remove pairs where the ratio of tokens between the source and target sentences is greater than 4.
- **Pair Length Ratio** – We remove pairs where the ratio between the string lengths of the source and target sentences is greater than 6.

Embedding-based Finally, we experiment with the use of sentence embedding models to compute embedding-based similarity between a sentence pair. We use LaBSE (Feng et al., 2020) models to embed both the source and target sentences then compute a cosine similarity score between the two. The pair must have a similarity score $0.7 \leq s \leq 0.96$ to be kept.

Statistics on the original and filtered corpus can be found on Table 1.

After preprocessing the parallel data, we learn a shared BPE (Sennrich et al., 2015b) vocabulary using SentencePiece[3] (Kudo and Richardson, 2018) with 32,000 units. All models in this paper use the same shared vocabulary.

2.2 Model Architecture

We experiment with two model sizes for each language pair: a **`Base`** model with 65M parameters and a **`Large`** model with 200M parameters. Both models use the standard Transformer (Vaswani et al., 2017) sequence-to-sequence architecture and are trained using Fairseq (Ott et al., 2019) with the hyperparameters listed in Table 2.

Training Hyperparameters	
Parameters	65M and 200M
Vocab Size	32,000
Tied Weights	Yes
Dropout	0.3
Attention Dropout	0.1
Weight Decay	0.0
Label Smoothing	0.1
Optimizer	Adam
Adam Betas	β_1=0.90, β_2=0.98
Adam ϵ	ϵ=1e-6
Learning Rate	7e-4
Warmup Steps	4,000
Total Steps	1,000,000
Batch size	64,000 tokens

Table 2: Hyperparameters used during training. When reporting model sizes, **`Base`** refers to 65M parameters, while **`Large`** refers to 200M.

We parallelize with 8 NVIDIA Tesla P100 GPUs and initially train for a total of 100K steps for experimentation. For the submitted systems trained with backtranslated data, we train for a total of 1M steps.

2.3 Backtranslation

We use backtranslation (Sennrich et al., 2015a) as a form of data augmentation to improve our initial models. We generate synthetic data via combined top-k and nucleus sampling:

$$\sum_{i=0}^{\delta_k} P(\hat{y}_i^{(T)}|x; \hat{y}^{(T-1)}) * \delta_{temp} \leq \delta_p \quad (1)$$

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

Backtranslation Hyperparameters	
Top-k (δ_k)	50
Top-p (δ_p)	0.93
Temperature (δ_t)	0.7
Beam	1.0
Length Penalty	1.0

Table 3: Hyperparameters used during backtranslation.

where δ_k is the top values considered for top-k sampling, δ_{temp} is the temperature hyperparameter, and δ_p is the maximum total probability for nucleus sampling.

Backtranslation is only performed once using the provided monolingual data. We produce a total of 10,000,000 synthetic sentences for the en→he direction and 73,278,018 synthetic sentences for the he→en direction. The same data preprocessing used on the original parallel corpus is then applied to the synthetic corpus. We produce backtranslations using **Large** 100K models with the sampling hyperparameters listed in Table 3.

Statistics on generated synthetic data before and after filtering can be found on Table 1.

2.4 Noisy Channel Reranking

We further improve translations by using Noisy Channel Reranking (Yee et al., 2019), which reranks every candidate translation token $\hat{y}_i^{(T)}$ using Bayes' Rule, as follows:

$$P(\hat{y}_i^{(T)}|x;\hat{y}^{(T-1)}) = \frac{P(x|\hat{y}^{(T-1)})P(\hat{y}^{(T-1)})}{P(x)} \quad (2)$$

where $P(\hat{y}_i^{(T)})$ refers to the probability of the ith candidate token at timestep T given source sentence x and current translated tokens $\hat{y}^{(T-1)}$.

All probabilities are parameterized as standard encode-decoder Transformer neural networks: the **Direct Model** $f_{\phi_D}(x,\hat{y}^{(T-1)})$ models $P(\hat{y}_i^{(T)}|x;\hat{y}^{(T-1)})$ or translation between source to target language; the **Channel Model** $f_{\phi_C}(x|\hat{y}^{(T-1)})$ models $P(x|\hat{y}^{(T-1)})$, or the probability of the target translating back into the predicted translation, and; the **Language Model** $f_{\phi_L}(\hat{y}^{(T-1)})$ models $P(\hat{y}^{(T-1)})$ or the probability of the translated sentence to exist. $P(x)$ is generally not modeled since it is constant for all y. This allows us to leverage a strong language model to guide the outputs of the direct model, while using a channel model to constrain the preferred outputs of the language model (which may be unrelated to the source sentence).

Decoding Hyperparameters	
Beam	5
Length Penalty	1.0
k2	5
CM Top-k	500
δ_{ch} en→he	0.2297
δ_{lm} en→he	0.2056
δ_{ch} he→en	0.2998
δ_{lm} he→en	0.2594

Table 4: Hyperparameters used for the final submission models. The values listed for δ_{ch} and δ_{lm} are the ones used for the final submission models. For testing with **Large 100K** models, we set both δ_{ch} and δ_{lm} to 0.3. "k2" refers to the number of candidates sampled per beam while "CM Top-k" refers to the number of most frequent tokens in the channel model's vocabulary that is used as its output vocabulary during decoding to save space.

During beam search decoding, we rescore the top candidates using the following linear combination of all three models:

$$P(\hat{y}_i^{(T)}|x;\hat{y}^{(T-1)})' = \frac{1}{t}log(P(x|\hat{y}^{(T-1)}) + \frac{1}{s}[\delta_{ch}log(P(x|\hat{y}^{(T-1)}) + \delta_{lm}log(P(\hat{y}^{(T-1)}))] \quad (3)$$

where s and t are source / target debiasing terms, δ_{ch} refers to the weight of the channel model, and δ_{lm} refers to the weight of the language model.

For Noisy Channel Reranking, our direct and channel models use the same size and setup at all times (i.e. if the direct model is a **Large** model trained for 100K steps, then the channel model is also a **Large** model trained for 100K steps in the opposite translation direction).

For the language model, we train one **Base**-sized decoder-only Transformer language model for English and one for Hebrew. We concatenate the cleaned data from the parallel corpus with the provided monolingual data for each language to train the LM. We use the same training setup as with translation models, except we use a weight decay of 0.01 and a learning rate of 5e-4.

Hyperparameters used for decoding with Noisy Channel Reranking can be found in Table 4.

2.5 Evaluation

We evaluate our models using two metrics: BLEU (Papineni et al., 2002) and ChrF++ (Popović, 2015), both scored via SacreBLEU[4] (Post, 2018). We develop our models using both the FLORES 200 (Costa-jussà et al., 2022) and NTREX 128 (Federmann et al., 2022) datasets, using the validation sets during training and reporting scores on the test sets.

To benchmark our models' performance, we mainly compare BLEU and ChrF++ against two (unconstrained) models: **`mBART 50 M2M`** (Tang et al., 2020), a 610M-parameter finetuned version of **`mBART`** for many-to-many translation, and **`NLLB 200 MoE`** (Costa-jussà et al., 2022), the full 54.5B-parameter mixture-of-experts version of **`NLLB 200`** for many-to-many translation.

2.6 Hyperparameter Search

To find the best values for δ_{ch} and δ_{lm}, as well as to understand how these parameters affect performance, we use Bayesian Hyperparameter Search. We use the **`Large`** 1M + BT models and run 1000 iterations of search, keeping the length penalty static at 1.0, and sampling both δ_{ch} and δ_{lm} from a gaussian with minimum of 0.01 and maximum of 0.99.

We perform this for both en→he and he→en translation directions and use the results for the final submission model.

3 Results

A summary of our results on benchmarks can be found on Table 5.

3.1 Benchmarking Results

Our submission systems (**`Large` 1M + BT + NC**) exhibit strong performance on both translation directions. On FLORES-200, we achieve 44.24 BLEU for en→he and 42.42 BLEU for he→en. The same systems score 33.77 BLEU for en→he and 36.89 BLEU for he→en on NTREX-128.

We note that these systems perform strongly when compared against much larger, unconstrained baseline models. On FLORES-200, we significantly outperform **`mBART 50 M2M`** on en→he by +24.75 BLEU and on he→en by +11.92 BLEU despite having 67% less parameters (200M vs 610M). Notably, our system performs only slightly worse compared to **`NLLB 200 MoE`** despite having **96% less parameters** compared to the mixture-of-experts model. On FLORES-200, we perform -2.56 BLEU worse on en→he and -6.58 BLEU worse on he→en compared to **`NLLB 200 MoE`**.

3.2 Hyperparameter Search Results

In order to find optimal hyperparameters for both δ_{ch} and δ_{lm}, we ran bayesian hyperparameter search for both at the same time while keeping length penalty static. We plot the results of the hyperparameter search over 1000 iterations in Figure 1.

We observe that performance is optimal when both hyperparameters are set to 0.2∼0.3, making performance increasingly worse as both hyperparameters approach closer to 1. We hypothesize that this signifies the model capturing the original distribution close enough that it does not need much correction or aid from the accompanying language model. Noisy channel reranking, however, is still empirically shown to be useful in this case as guidance from the language model produces better candidates in cases where the direct model may be searching a too-constrained space.

3.3 Ablations

We explored multiple configurations of our submission systems in terms of model size, presence of synthetic data during training, and the use of reranking methods during online decoding. Our results show that each step improves performance directly:

- The initial **`Base` 100K** performs at 39.88 BLEU for en→he on FLORES-200.
- Increasing the size to 200M parameters (**`Large` 100K**) improves performance by +1.38 BLEU.
- Adding backtranslated data (**`Large` 100K + BT**) is by far the most beneficial, improving performance by +2.06 BLEU.
- We then experiment with longer training times (1M iterations for **`Large` 1M + BT**) to adapt to the new dataset size, increasing the score by +0.44 BLEU.
- Finally, using noisy channel reranking (**`Large` 1M + BT + NC**) improves the score by +0.48 BLEU.

[4]SacreBLEU outputs the following signature for evaluation: nrefs:1|case:mixed|eff:no|tok:spm-flores|smooth:exp|version:2.2.1

	FLORES-200				NTREX-128			
	EN → HE		HE → EN		EN → HE		HE → EN	
Model	**BLEU**	**ChrF++**	**BLEU**	**ChrF++**	**BLEU**	**ChrF++**	**BLEU**	**ChrF++**
Base 100K	39.88	56.34	12.06	29.46	31.47	48.32	29.85	52.53
Base 100K + NC	40.22	56.55	38.75	60.52	32.10	48.93	31.86	54.57
Base 100K + BT	41.50	57.46	38.73	60.80	31.27	47.90	34.09	56.10
Base 100K + BT + NC	41.66	57.59	40.43	62.17	32.05	48.62	35.76	57.65
Large 100K	41.26	57.46	39.07	60.06	32.49	48.95	31.08	53.19
Large 100K + NC	41.46	57.64	40.53	61.49	32.80	49.34	33.12	55.16
Large 100K + BT	43.32	58.62	40.91	61.58	32.90	49.11	35.48	56.04
Large 100K + BT + NC	43.26	58.72	41.92	62.64	33.18	49.42	36.79	57.37
Large 1M + BT	43.76	58.29	41.00	61.16	33.35	49.22	35.83	56.02
Large 1M + BT + NC	**44.24**	**59.36**	**42.42**	**62.21**	**33.77**	**49.69**	**36.89**	**56.92**
mBART50 M2M (610M)	19.49	46.7	30.50	55.00	14.80	42.30	27.02	51.21
NLLB 200 MoE (54.5B)	46.80	59.80	49.00	67.40	-	-	-	-

Table 5: Compiled results for all experiments. "BT" refers to the model being trained with backtranslated data in addition to original filtered data. "NC" refers to the use of Noisy Channel Reranking. Evaluation scores for **NLLB 200 MoE** are taken from its official published scores for FLORES-200. We fail to report independent NTREX-128 scores for **NLLB 200 MoE** due to a lack of computational resources.

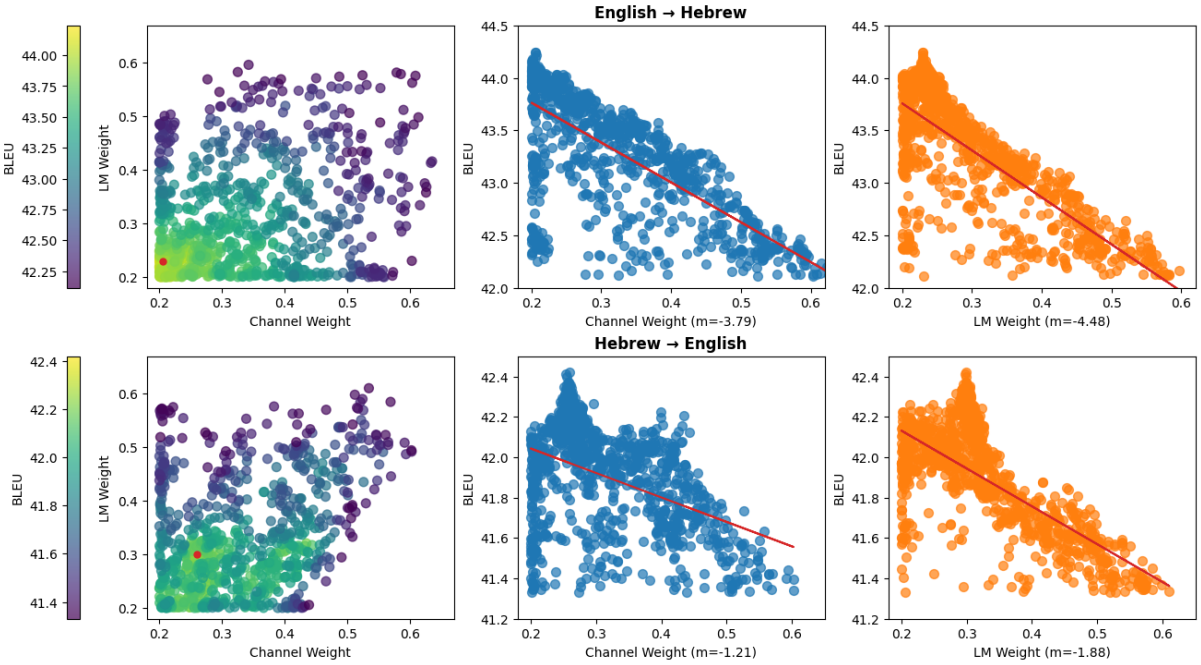

Figure 1: Bayesian hyperparameter search results for δ_{ch} and δ_{lm} while keeping constant length penalty. The leftmost column shows BLEU score against both δ_{ch} and δ_{lm} with the best performing model (**Large** 1M + BT + NC) plotted in red. The middle and rightmost columns show δ_{ch} and δ_{lm} against BLEU, respectively, with their respective regression lines (in red) and regression coefficient (m) in the caption.

Overall, all of our methods improve performance by a total of 4.36 BLEU for the en→he direction on FLORES-200.

We note an interesting jump in performance from **Base 100K** to **Large 1M + BT + NC** on the FLORES-200 he→en direction at +30.36 BLEU. **Base 100K** underperforms at 12.06 BLEU, and we hypothesize that this is due to the model not having enough capacity to embed information from Hebrew, which causes it to greatly benefit from the guidance of a language model during noisy channel reranking.

4 Conclusion

In this paper, we describe our submissions to the WMT 2023 General Translation Task. We participate in two constrained tracks: en→he and he→en.

We submit two monodirectional models based on the Transformer architecture. Both models are trained using a mix of original and synthetic back-translated data, filtered and curated using a comprehensive data processing pipeline that combines embedding-based, heuristic-based, and ratio-based filters. Additionally, we employ noisy channel reranking to improve translation candidates using a language model and a channel model trained in the opposite direction.

On two benchmark datasets, our systems outperform **mBART50 M2M** and perform slightly worse than **NLLB 200 MoE**, both unconstrained systems with significantly more parameters.

Our results show that established best practices still perform strongly on constrained systems without the need for extraneous data sources as is with unconstrained systems for the same translation directions.

Limitations

We benchmark on datasets that are publicly available with permissive licenses for research.

We note that we are unable to study scale properly for translation models due to a lack of stronger compute resources. The same constraint also prevents us from training multiple iterations of the same model with differing random seeds. Our systems' true performance may thus be higher or lower depending on the machine random state at the start of training time.

Lastly, our models are trained on Hebrew, which is a language that we do not speak. We are therefore unable to manually evaluate if the output translations are correct, natural, or semantically sound.

Ethical Considerations

Our paper replicates best practices in data preprocessing, model training, and online decoding for translation models. Within our study, we aim to create experiments that replicate prior work under comparable experimental conditions to ensure fairness in benchmarking.

Given that we do not speak the target language in the paper, we report performance in comparison to other existing models. We do not claim that "strong" performance in a computational setting correlates with good translations from a human perspective.

Lastly, while we do not use human annotators for this paper, the conference (WMT) itself does for human evaluations on the General Translation Task. We disclose this fact and note that annotations (and therefore scores) may be different across many speakers of Hebrew.

References

Dan Bareket and Reut Tsarfaty. 2021. Neural Modeling for Named Entities and Morphology (NEMO2). *Transactions of the Association for Computational Linguistics*, 9:909–928.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Jan Christian Blaise Cruz and Charibeth Cheng. 2021. Improving large-scale language models and resources for filipino. *arXiv preprint arXiv:2111.06053*.

Jan Christian Blaise Cruz and Lintang Sutawika. 2022. Samsung research philippines-datasaur ai's submission for the wmt22 large scale multilingual translation task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1034–1038.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Lintang Sutawika and Jan Christian Blaise Cruz. 2021. Data processing matters: Srph-konvergen ai's machine translation system for wmt'21. *arXiv preprint arXiv:2111.10513*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.