Samsung R&D Institute Philippines @ WMT 2024 Low-resource Languages of Spain Shared Task

Dan John Velasco[★,a] **Manuel Antonio Rufino**[★,a] **Jan Christian Blaise Cruz**[a,b]
[a]Samsung R&D Institute Philippines [b]MBZUAI
{dj.velasco,ma.rufino}@samsung.com, jan.cruz@mbzuai.ac.ae
[★]**Equal Contribution**

Abstract

This paper details the submission of Samsung R&D Institute Philippines (SRPH) Language Intelligence Team (LIT) to the WMT 2024 Low-resource Languages of Spain shared task. We trained translation models for Spanish to Aragonese, Spanish to Aranese/Occitan, and Spanish to Asturian using a standard sequence-to-sequence Transformer architecture, augmenting it with a noisy-channel reranking strategy to select better outputs during decoding. For Spanish to Asturian translation, our method reaches comparable BLEU scores to a strong commercial baseline translation system using only constrained data, backtranslations, noisy channel reranking, and a shared vocabulary spanning all four languages.

1 Introduction

This paper details our constrained system for translating from Spanish to Aragonese (spa→arg), Aranese/Occitan (spa→arn), and Asturian (spa→ast) for the WMT24 Shared Task: Translation into Low-Resource Languages of Spain. We trained standard sequence-to-sequence Transformer architecture (Vaswani et al., 2017) from scratch combined with heavy data preprocessing (Cruz, 2023), data augmentation via backtranslation (Sennrich et al., 2016a), and noisy channel reranking (Yee et al., 2019) to achieve performance that is comparable to Apertium (Khanna et al., 2021) v3.9.6 for spa→ast. We present ablation results of the effect of data augmentation via backtranslation and noisy channel reranking with respect to BLEU scores. Furthermore, we analyzed the generated translations and we found that the model learned to regurgitate, i.e. repeat with minor modifications, the source Spanish sentences for the spa→ast case. We also identify rarely occurring characters that the model failed to learn. Lastly, we also investigated the effect of the length of backtranslated data on improving model performance.

2 Methodology

2.1 Environment

For preprocessing, training, and generation, we used fairseq 0.12.2 and PyTorch 1.12.1. The training was done on either 2x NVIDIA Quadro GPUs or 8x NVIDIA P100 GPUs. We used Apertium[1] v3.9.6 for generating baseline results and generating backtranslated (BT) data whenever available for the language pair.

2.2 Data Preprocessing

We trained on the OPUS dataset (Tiedemann, 2016) for all language pairs. The data preprocessing pipeline utilizes the ratio-based filters and embedding-based filters of Samsung R&D Institute Philippines' WMT23 entry (Cruz, 2023). The dataset statistics before and after preprocessing can be found in Table 1.

For the parallel data, the data preprocessing pipeline are as follows: remove exact duplicate parallel data → ratio-based filters → embeddings-based filters. The ratio-based filters remove sentences based on sentence length, token length, character to token ratio, pair token ratio, and pair length ratio. Exact details on these criteria are explained in (Cruz, 2023). Similar to last year's paper, we tokenized and detokenized sentences using SacreMoses[2] before and after running our filters, respectively. The embeddings-based filter filters data based on the cosine similarity of a sentence pair using LaBSE (Feng et al., 2022). Using the methodology of (Cruz, 2023), pairs with a cosine similarity $0.7 \leq s \leq 0.96$ are kept.

For monolingual data, we combined the monolingual data of the target language and the target

[b]Work done while at Samsung R&D Institute Philippines

[1]https://wiki.apertium.org/wiki/Install_Apertium_core_using_packaging

[2]https://github.com/alvations/sacremoses

Proceedings of the Ninth Conference on Machine Translation (WMT), pages 892–900
November 15–16, 2024. ©2024 Association for Computational Linguistics

source→target	Pairs	Words (source)	Words (target)	% Kept
spa→arg	58,284	746,567	733,985	100
spa→arg Filtered	21,362	181,523	190,724	36.6
spa→arg Filtered + BT†	81,195	849,031	857,111	-
spa→ast	13,393,052	310,197,263	298,687,582	100
spa→ast Filtered	620,168	6,495,284	6,442,051	4.6
spa→ast Filtered + BT	920,168	11,758,510	11,629,822	-
spa→arn	2,032,440	14,046,448	13,891,193	100
spa→arn Filtered	779,615	4,807,268	5,020,187	38.4
spa→arn Filtered + BT†	1,079,615	8,835,974	9,040,705	-

Table 1: Statistics of OPUS parallel data before and after filtering and the addition of backtranslated data (BT). The % Kept is the percentage of pairs left after filtering ("-" means not applicable). † means BT data was generated via Apertium.

side of parallel data from OPUS and then removed exact duplicates. We used this monolingual data to train language models for each target language.

After preprocessing of parallel and monolingual data, we apply train and validation split of 95% and 5%, respectively.

Lastly, for the training corpus of the tokenizer, we combined the filtered parallel data of all three language pairs. We used this combined data to learn a shared BPE (Sennrich et al., 2016b) vocabulary that spans Spanish, Aragonese, Aranese, and Asturian consisting of 31,960 tokens using SentencePiece (Kudo and Richardson, 2018). This shared vocabulary was used by all models for generating submissions to WMT24. We used this approach as the four languages belong to the same language family.

2.3 Augmenting Data with Backtranslation

We augmented the filtered training data using backtranslation (Sennrich et al., 2016a). For each language pairs for both source→target (except spa→ast) and target→source directions, Apertium 3.9.6 was used to generate BT data. Due to the lack of direct translation support for some language pairs in Apertium, the translation for arg→spa went through the following translation path: Aragonese → Catalan → Interlingua → Spanish[3]. For arn→spa, it goes through Aranese → Catalan → Spanish[4].

Translation from Asturian to Spanish is not supported by Apertium. Alternatively, we used the ast→spa model that was originally intended for noisy channel reranking (NCR), a technique which will be explained in Section 2.5, to generate BT data. For decoding, we used combined top-k and nucleus sampling:

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

where δ_k is the top values considered for top-k sampling, δ_{temp} is temperature, δ_p is the maximum total probability for nucleus sampling. For these hyperparameters, we used the same values as (Cruz, 2023) which are as follows: δ_k = 50, δ_{temp} = 0.7, and δ_p = 0.93.

Once the BT data for each language pairs and translation direction are generated we took a subset in different ways. For BT data for training Direct Translation Models (spa→arg/ast/arn), we used all the generated BT data for spa→arg since it's less than 300K. For BT data of spa→ast and spa→arn, we keep the longest 300K sentences.

For BT data for training Channel Translation Models (arg/ast/arn→spa), we used all the BT data for arg→spa since it's less than 100K. For ast→spa, we randomly sampled 100K sentences. Due to time constraints, we did not generate BT data for arn→spa.

2.4 Model Training

For each language pair, we trained three types of models: a **Direct Translation Model**, a **Channel Model**, and a **Language Model** which will be detailed in the following subsections. These three models will be combined via Noisy Channel Reranking (Yee et al., 2019) which will be explained in Section 2.5.

2.4.1 Direct Translation Models

For each direct translation models (spa→arg/arn/ast), we trained encoder-decoder

[3]Apertium language codes: arg-cat→cat-ina→ina-spa
[4]Apertium language codes: oc_aran-ca→cat-spa

Transformer architecture (Vaswani et al., 2017) from scratch with and without BT data. We used the large variant of transformers which has 213M parameters[5]. We describe two training configurations: **tf-large60k** which was trained for 60,000 steps of which 3,000 are warmup steps, and **tf-large100k** which was trained for 100,000 steps of which 10,000 are warmup steps. Training settings with "-plusbt" suffix indicates that the model was trained on a mixture of provided training data and BT data. Otherwise, it indicates the model is trained only on the provided training data. For example, **tf-large100k-plusbt** means the model was trained on the mixture of provided training data and BT data for 100,000 steps of which 10,000 are warm up steps.

For both settings and all language directions, unless stated otherwise, we used the same hyperparameters in Table 2. For generating WMT24 submissions, we used models trained on **tf-large100k-plusbt** setting as our Direct Translation Model.

2.4.2 Channel Translation Models

For the channel translation models (arg/arn/ast→spa), we used the same architecture and hyperparameters as the direct translation models, except it was trained on **tf-large60k-plusbt** setting, batch size/max tokens of 10,000, and learning rate of 7e-4 (arg→spa and arn→spa) and 5e-5 (ast→spa). These were used as channel models for noisy channel reranking which is explained further in Section 2.5 and for performing hyperparameter sweeps of noisy channel reranking parameters detailed in Section 2.6.

2.4.3 Language Models

We trained monolingual language models for Aragonese, Aranese, and Asturian from scratch using the decoder-only part of the original Transformer architecture as described in (Vaswani et al., 2017). We used the base variant which has 65M parameters[6]. For all languages, we used Adam optimizer (Kingma and Ba, 2017) with β_1=0.90, β_2=0.98. We trained for a maximum of 250,000 steps of which 4,000 are warmup steps. The warmup initial learning rate is 1e-7 and the max learning rate is 5e-4 and then decayed following an Inverse Square root learning rate schedule. The batch size / max tokens is 40,000, and the dropout is 0.1. These models were used in noisy channel reranking which is explained further in Section 2.5 and for performing hyperparameter sweeps of noisy channel reranking parameters detailed in Section 2.6.

[5]Fairseq model code: transformer_wmt_en_de_big

[6]Fairseq model code: transformer_lm

Training Hyperparameters	
Vocab Size	31,960
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	5e-5
LR Schedule	Inverse Sqrt
Batch Size	8,000 tokens

Table 2: Fixed hyperparameters for direct translation models.

2.5 Noisy-Channel Reranking

Similar to (Cruz, 2023), we experimented with using Noisy Channel Reranking (Yee et al., 2019) to improve translations. This works by using a direct translation model (source→target), channel model (target→source) and a monolingual language model (target only) to rescore every candidate translation token during beam search decoding. The score of the candidate translation token $\hat{y}_i^{(T)}$ at time step T is recomputed using a linear combination of all three models:

$$P(\hat{y}_i^{(T)}|x;\hat{y}^{(T-1)})' = \frac{1}{t}\log(P(y|\hat{x}^{(T-1)}) + \frac{1}{s}[\delta_{ch}\log(P(x|\hat{y}^{(T-1)}) + \delta_{lm}\log(P(\hat{y}^{(T-1)}))] \quad (2)$$

where t is the length of target sentence y and s is the length of source sentence x which serves as debiasing terms. The δ_{ch} and δ_{lm} are weights of the channel model and language model, respectively, which controls the influence of the models to the final score. For this paper, both δ_{ch} and δ_{lm} were set to 0.5

2.6 Hyperparameter Sweeping

Similar to (Cruz, 2023), we utilized a Bayesian hyperparameter search to find an optimal value for

Setting	BLEU					
	FLORES+ dev			WMT24 Test		
	spa→arg	spa→ast	spa→arn	spa→arg	spa→ast	spa→arn
Apertium (baseline)	**70.3**	22.6	**42.4**	-	-	-
No BT; No NCR	18.3	23.9	8.7	13.4	16.8	7.2
No BT; w/ NCR	21.3	24.0	8.7	16.5	16.9	7.2
w/ BT; No NCR	35.4	**24.4**	14.4	26.7	**17.5**	**7.7**
w/ BT; w/ NCR	37.1	24.3	13.7	**28.2**	17.2	7.2

Table 3: BLEU scores of various system configurations compared to Apertium. BT and NCR denotes backtranslated data and noisy channel reranking, respectively. Highest score per language pair are in bold.

Model configuration	BLEU
tf-base100k w/o NCR	19.3
tf-base100k w/ NCR	20.7
tf-base100k-plusbt w/o NCR	36.4
tf-base100k-plusbt w/ NCR	**37.6**
tf-large100k w/o NCR	18.3
tf-large100k w/ NCR	21.3
tf-large100k-plusbt w/o NCR	35.4
tf-large100k-plusbt w/ NCR	**37.1**

Table 4: Ablation results for spa→arg. NCR denotes noisy channel reranking.

Setting	BLEU		
	whole	mid	long
no-BT (baseline)	8.2	8.4	8.2
short-BT	10.5	9.6	10.4
mid-BT	11.6	10.7	11.6
long-BT	**14.3**	**11.4**	**14.3**

Table 5: BLEU scores per length group of BT data. long-BT outperforms all other settings in all test setups.

length penalty. The length penalty sweep was performed for 137 iterations sampling from a uniform distribution with minimum 0.0 and maximum 2.0. Hyperparameter sweeping was performed using the **tf-large60k-plusbt** direct translation models with noisy channel reranking enabled on the Spanish to Aragonese language pair. Translations for the hyperparameter sweep were generated from the copy of FLORES+ (Team et al., 2022) found in the PILAR (Galiano-Jiménez et al., 2024) repository[7]. The results of this sweep were used on all language pairs. We performed the sweep on spa→arg only and on a **tf-large60k-plusbt** model due to hardware and time constraints. Our sweeps showed that setting length penalty to 1.726 is optimal.

3 Results and Discussion

In this section, we discuss the results of our experiments and discuss our findings. Experiments were performed using the copy of FLORES+ (Team et al., 2022) found in the PILAR (Galiano-Jiménez et al., 2024) repository were computed using SacreBLEU[8] (Post, 2018).

[7]https://github.com/transducens/PILAR

[8]SacreBLEU signature: nrefs:1|case:mixed|eff:no|tok:flores101|smooth:exp|version:2.4.2

For all translations, we used the following decoding hyperparameters: top_k=50, top_p= 0.93, temperature=0.7, beam=5. Additional hyperparameters are specified per experiment.

3.1 Comparison Against Baselines

We compare our system against Apertium 3.9.6. Results are listed in Table 3. We observe that Apertium yields the highest BLEU score for spa→arg and spa→arn. For spa→ast, the systems trained with BT data both outperform the Apertium baseline.

Our method performs worst on the spa→arn language pair while it performs best for spa→arg. However for both of these pairs our system is outperformed by Apertium. From this we can conclude that our current pipeline cannot overcome the low resource nature of these language pairs in order to close the gap with Apertium. For spa→ast, we were able to outperform Apertium with a difference of 1.8 BLEU.

3.2 Ablations

We perform an ablation study by varying model size, use of BT data, and use of noisy channel reranking. Due to hardware and time constraints, we only perform our ablations in the spa→arg direction. Results are summarized in Table 4.

We observe that the addition of BT data and

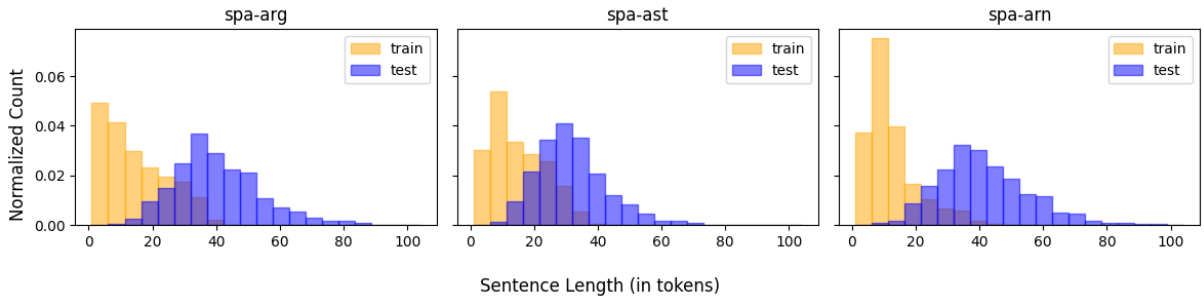


Figure 1: Sequence length distribution of the target side of the train (filtered) and test set per language pair. spa-arn has the least overlap between train and test and has the most short examples in training data which hints why the BLEU score is relatively lower compared to other language pairs.

noisy channel reranking resulted in an increased BLEU score. Using both strategies yields the highest BLEU score for both base and large model sizes. It is notable that the base model with both BT data and noisy channel reranking yields the highest BLEU score in our ablation study. We speculate that this be due to the large model having too many parameters for the given task or a lack of data. Another reason is that spa-arn is relatively easier compared to other language pairs because the source and target are more similar with each other as shown in Table 6. More experiments are needed to confirm these.

3.3 Adding Longer Examples Improves BLEU Better than Shorter Examples

For the spa→arn baseline model (No BT), we observed a BLEU score of 8.7 on FLORES+ dev set. One possible explanation for the low score is the mismatch between the length distribution of training and test data. We observed that the training data is comprised mostly of short examples while the FLORES+ dev set is relatively longer (see Figure 1). We hypothesize that adding longer examples to the training set will improve BLEU score, especially on longer examples.

To provide evidence for the hypothesis, we generated BT data of size 100,000 for different length groups namely, short-BT (1-10 words), mid-BT (11-20 words), long-BT (20+ words). We mixed the BT data with the training data then trained a model for each setup. We trained each model for 50,000 of which 5,000 are warmup steps. We used the same training hyperparameters as in Table 2. For fair comparison, we trained a baseline model (no-BT) using the same training hyperparameters. For generating the translations, we did not use noisy channel reranking and we fix the length penalty to 1.0. The results are summarized in Table 5.

The result shows that long-BT gives an absolute BLEU score improvement of +6.1 over baseline, followed by mid-BT (+3.4), and then short-BT (+2.3). This tells us that while augmenting with BT data generally improves the performance, strategically adding more long examples can give the most improvements in a resource-constrained setting. To strengthen this claim further, we performed a fine-grained test by grouping FLORES+ dev set by length groups (mid/long). For this experiment, we did not include the short length group because it only contains 3 examples after grouping. The results shows that long-BT gives the most improvements on mid and long test groups, followed by mid-BT and short-BT (see Table 5). This suggests that training on longer sequences also improves performance on shorter sequences.

While this experiment shows empirical results that adding longer examples improves the overall BLEU score better than adding shorter examples, it does not say something about the quality and diversity of the text. It is possible that these findings might not hold if the long examples are of low quality. Another possible explanation on why long-BT outperforms its shorter counterparts is because, with the same number of examples of 100,000, long-BT contains more tokens than short-BT and mid-BT. To further solidify the claim that adding longer examples improves the overall BLEU score better than adding shorter examples, more experiments are needed where total token count per length group are equal or close to each other.

Language Pair	JS (generated) ↑	ED (generated) ↓	JS (ground truth) ↑	ED (ground truth) ↓
spa→arg	0.34	0.59	**0.34**	**0.57**
spa→arn	0.23	0.69	0.13	0.83
spa→ast	**0.44**	**0.44**	0.23	0.76

Table 6: Average Jaccard similarity (JS) and average normalized edit distance (ED) between source and generated translations and ground truth translations. Results confirm our observatoin that our system is regurgitating Spanish source sentences in the spa→ast direction. Results also suggest that the Spanish and Aragonese sentences in the FLORES+ dev set are more similar to each other compared to others.

3.4 Regurgitation of Spanish Sentences in Generated Translations

We observed that our model was producing some translations that were only slightly altered versions of the source Spanish sentence. To empirically evaluate the extent of this problem for our system, we compare the BPE tokenized source Spanish sentences of the FLORES+ dataset from PILAR to the corresponding generated translations made by our system and the corresponding ground truth. We compared this system via two metrics: Jaccard similarity (JS) and normalized edit distance (ED). To compute the two metrics between two BPE encoded sentences S_1 and S_2, we get the set of tokens of each sentence T_1 and T_2 and compute Jaccard similarity as

$$JS = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

and normalized edit distance as

$$ED = \frac{D(S_1, S_2)}{\max(|S_1|, |S_2|)}$$

where $D(S_1, S_2)$ denotes token-level Levenshtein distance between BPE encoded sentences S_1 and S_2. We divide by the maximum length between S_1 and S_2 to ensure that we get a value between 0 and 1. Results of this analysis are summarized in Table 6.

We observe that spa→ast shows the highest average Jaccard similarity and the lowest normalized edit distance among language pairs for generated translations; however, the corresponding metrics for the spa→ast ground truth translations tell a different story. Ground truth translations for spa→ast show a lower Jaccard similarity and a higher normalized edit distance, indicating that we may be regurgitating Spanish sentences.

Below is a sample of a Spanish sentence together with a generated Asturian translation which exhibits regurgitation and the corresponding ground truth translation. Notice how the generated translation is closer in similarity to the Spanish sentence than the correct Asturian translation. In the below example, "S -" is the source spanish sentence, "H -" is the generated Asturian translation, "T -" is the ground truth Asturian translation, "J -" is the jaccard similarity compared to the source Spanish sentence, and "E -" is the normalized edit distance compared to the source Spanish sentence. All sentences are BPE encoded.

```
S - _Apenas _pas adas _las _11: 00 _h , _los
    _integrantes _de _la _manifestación _bloque
    aron _la _circulación _del _car ril _de
    _White h all _que _va _hacia _el _norte .

H - _Ap enes _pasa es _les _11: 00 _h , _los
    _integrantes _de _la _manifestación _blo qui
     aron _la _circulación _del _car ril _de
    _White h all _que _va _escontra ' l _norte .
J - 0.553
E - 0.303

T - _X usto _depués _de _les _11: 00, _los
    _manifestantes _blo qui aron _el _trá ficu
    _nel _sentíu _norte _en _White h all .
J - 0.244
E - 0.833
```

For spa→arg, Jaccard similarity and normalized edit distance are similar for both generated translations and ground truth translations. We note that this language pair has the highest Jaccard similarity and lowest normalized edit distance between its source Spanish sentences and ground truth Aragonese translations. This indicates that there is a degree of similarity between the Spanish and Aragonese sentences in the dataset which may explain why the spa→arg model exhibited the highest BLEU score in our baseline comparison. We provide a sample below where the source Spanish sentence is similar to the ground truth Aragonese translation.

```
S - _En _el _partido , _Nadal _acumul ó _un  _8
    8% _de _puntos _ne tos _y _ganó _76 _en _el
    _primer _servicio .

H - _En _o _parti to , _Nadal _acumul ó _un  _8
    8% _de _puntos _ne tos _y _ganó _76 _en _o
```

```
    _primer _servicio .
J - 0.792
E - 0.174

T - _En _o _parti u , _Nadal _acumul ó _un  _8
    8% _de _puntos _ne tos _y _ganó _76 _en _o
    _primer _servicio .
J - 0.792
E - 0.174
```

We plot the histogram of Jaccard similarity and normalized edit distance for all language pairs in Figures 2, 3, and 4.

3.5 Character Set Analysis

We observe that our generated translations do not contain all characters present in the ground truth as shown in Table 7. For all languages, the missing characters are present in the training data with the exception of Õ for Asturian and Aragonese. All missing characters constitute less than 1% of the training data which may explain why they were not learned by our models.

4 Conclusions

We detailed our constrained system for translating from Spanish to Aragonese (spa→arg), Aranese/Occitan (spa→arn), and Asturian (spa→ast). These systems were trained from scratch on constrained data, augmented by backtranslated (BT) data. Translations were further improved by utilizing Noisy Channel Reranking. This approach outperformed Apertium on the spa→ast translation direction. Our ablation study for spa→arg showed that utilizing backtranslation and noisy channel reranking improves BLEU score. However, more experiment is needed for other language pairs. Our ablation experiment also suggests that smaller models are capable enough for spa→arg, at least for this train and test set.

We investigated the cause of low BLEU score for spa→arn despite having more data (after filtering) than spa→arg and spa→ast. We linked it to the train-test mismatch of spa-ast data in terms of sequence length. We also found that adding longer backtranslated data improves overall BLEU score even in shorter sequences.

Lastly, we observed that our model for spa→ast was regurgitating Spanish sentences in Asturian translations and that characters with low frequencies in the training data are not being learned by our models.

Limitations

We are unable to evaluate whether the translations we generate are syntactically or semantically sound due to the fact that none of us speak Spanish, Aragonese, Asturian, or Aranese/Occitan.

References

Jan Christian Blaise Cruz. 2023. Samsung R&D institute Philippines at WMT 2023. In *Proceedings of the Eighth Conference on Machine Translation*, pages 103–109, Singapore. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Aarón Galiano-Jiménez, Felipe Sánchez-Martínez, and Juan Antonio Pérez-Ortiz. 2024. Pilar.

Tanmai Khanna, Jonathan N. Washington, Francis M. Tyers, Sevilay Bayatlı, Daniel G. Swanson, Tommi A. Pirinen, Irene Tang, and Hèctor Alòs i Font. 2021. Recent advances in apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine Translation*, 35(4):475–502.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

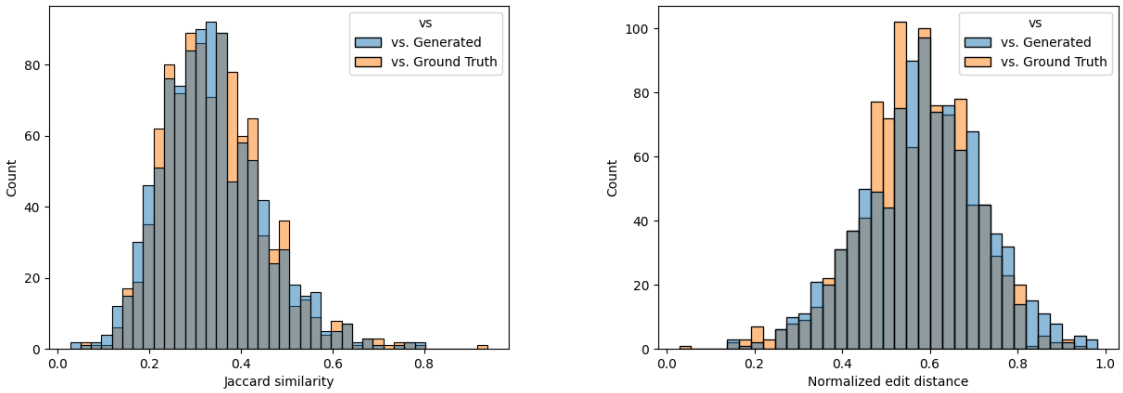
Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

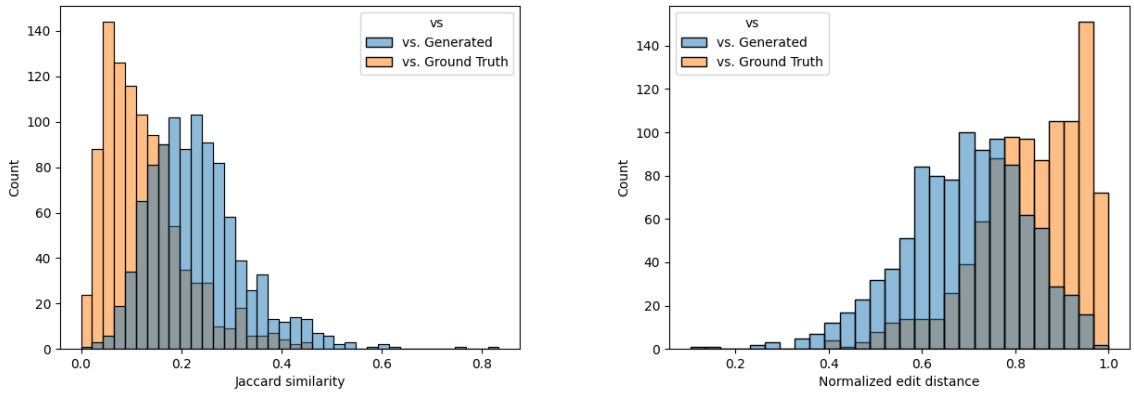
Language	Missing Characters	Frequency in Training Data	# Characters in Training Data
Aragonese	» «] & õ [Õ	1,583	4,713,907
Aranese/Occitan	& « » Ç Ò Õ *U+0301* ’	96,233	36,014,337
Asturian	" Õ Ú *U+1E24* ḥ – — ’	56,798	42,897,857

Table 7: Characters present in ground truth translations but missing in generated translations together with their frequency in training data compared to the total number of characters in training data. Unicode symbol code in italics listed when a character is unsupported by LaTeX. All missing characters constitute less that 1% of the training data.



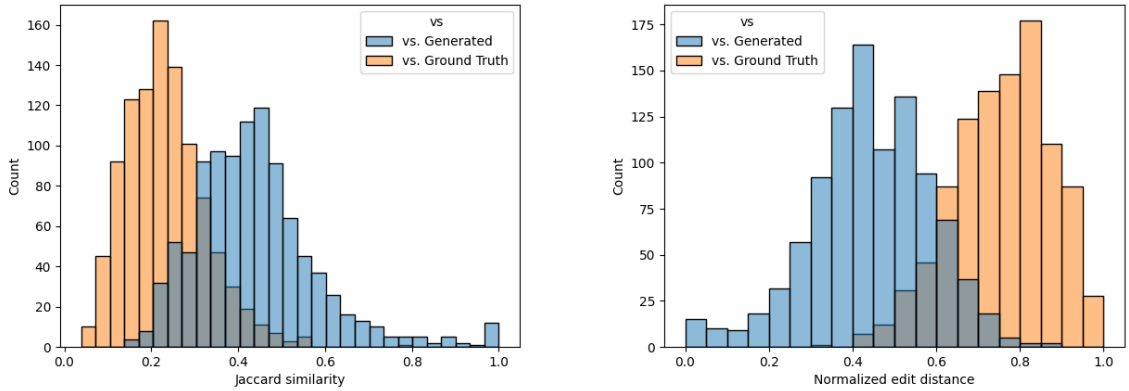
(a) (b)

Figure 2: Distribution of Jaccard similarity and normalized edit distance for spa→arg of source sentences vs generated translations and ground truth translations. We can see that the distributions for both Jaccard similarity and normalized edit distance almost entirely overlap. Taken together with the means from Table 6, these show that any regurgitation our model exhibits can also be seen in the ground truth test data.



(a) (b)

Figure 3: Distribution of Jaccard similarity and normalized edit distance for spa→arn of source sentences vs generated translations and ground truth translations. We can see in (a) that while Jaccard similarity of generated translations vs. source Spanish sentences is higher compared to that of ground truth translations vs. source Spanish sentences, they both tend to be less than 0.4. In (b), we see that while normalized of generated translations vs. source Spanish sentences is lower compared to that of ground truth translations vs. source Spanish sentences, they both tend to be greater than 0.6. This indicate low amounts of regurgitation in the case of our spa→arn system.



(a) (b)

Figure 4: Distribution of Jaccard similarity and normalized edit distance for spa→ast of source sentences vs generated translations and ground truth translations. We see in (a) that the Jaccard similarity of generated Asturian translations compared to source Spanish sentences is higher than that of ground truth translations compared to source sentences. In (b), we see that the normalized edit distance of generated translations compared to source sentences is lower than that of ground truth vs. source sentences. This indicates that our model is regurgitating more Spanish words rather than translating to Asturian.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Jörg Tiedemann. 2016. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.