Using Self-Improving With Transformers for MSLG-SPA Task

José Adrián Rodríguez González[1]

[1]University of Guanajuato Comunidad Palo Blanco, Carretera Salamanca-Valle de Santiago km 3.5+1.8. Salamanca, Gto., C.P. 36885, México

Abstract

Despite recent advances in Neural Machine Translation (NMT), significant challenges remain in low-resource scenarios, particularly for languages lacking standardized representations, such as Mexican Sign Language Gloss (MSLG) and Spanish. To address this issue, we evaluate a self-improving framework based on semi-supervised learning, synthetic data generation, and back-translation techniques. The approach is trained on a low-resource dataset and subsequently tested on an out-of-vocabulary (OOV) benchmark. While the proposed method achieved a BLEU score of 40.35 on in-domain data, its performance drops significantly to 3.36 BLEU on the MSLG2SPA benchmark. This discrepancy indicates that, although the framework effectively captures semantic and structural patterns present in the training data, it struggles to generalize to unseen lexical items and compositions. These findings highlight the key limitations of self-learning approaches in low-resource settings and emphasize the need for stronger lexical support mechanisms to reduce noise propagation during training and improve generalization capabilities.

Keywords

Neural Machine Translation, Self-Improving models, Transformers, Semi-supervised models, Neural Machine Translation, Mexican Sign Language Gloss

1. Introduction

Despite recent advances in neural machine translation (NMT), several challenges remain unsolved, particularly in low-resource scenarios where limited textual data is available [1]. Examples of such languages include Slovene, Zulu, Swahili among others. Addressing data scarcity requires the use of data augmentation, sampling strategies and other techniques during the training phase. A related challenge arises in the translation between Spanish and Mexican Sign Language Gloss [2][3]. Although Mexican Sign Language has been officially recognized in 2014 [4], available linguistic resources remain limited. Existing dictionaries provide mappings between MSLG and Spanish [5].

To address this gap, the IberLEF 2026 shared task [3] focuses on developing systems capable of translating from Spanish to MSLG and from MSLG to Spanish. This task highlights the challenges associated with low-resource conditions, including limited vocabulary coverage and the presence of out-of-vocabulary (OOV) terms

2. Dataset

The dataset consists of 490 samples, each containing a MSLG sentence and its corresponding Spanish translation. The MSLG data includes prefixes that encode linguist information such as the location, genre. and other grammatical features.

Due to its visual-spatial nature, MSLG does not follow the same syntactic and grammatical rules as Spanish. For example, the expression *VIVI-YO America* illustrates how gloss sequences may omit function words and differ in word order. Consequently,translation into Spanish requires reordering

IberLEF 2026, September 2026, León, Spain

ja.rodriguez.g@ugto.mx (J. A. R. González)

https://github.com/JoseAdrianRodriguezGonzalez (J. A. R. González)

0009-0009-7068-8852 (J. A. R. González)

tokens, removing gloss specific prefixes, and introducing appropriate prepositions, connectors, and language rules to produce fluent sentences.

The test data consists of 245 samples for each translation direction: MSLG-to-Spanish and Spanish-to-MSLG.

According the figure 1, most MSLG sentences in the training set, have a length of 4 followed by 5 words. whereas Spanish sentences are generally longer, with most samples containing 6 words to 7 words. This differences reflects the more compact representation of MSLG compared to Spanish, as it encodes meaning in a more condensed, visually grounded form [5],

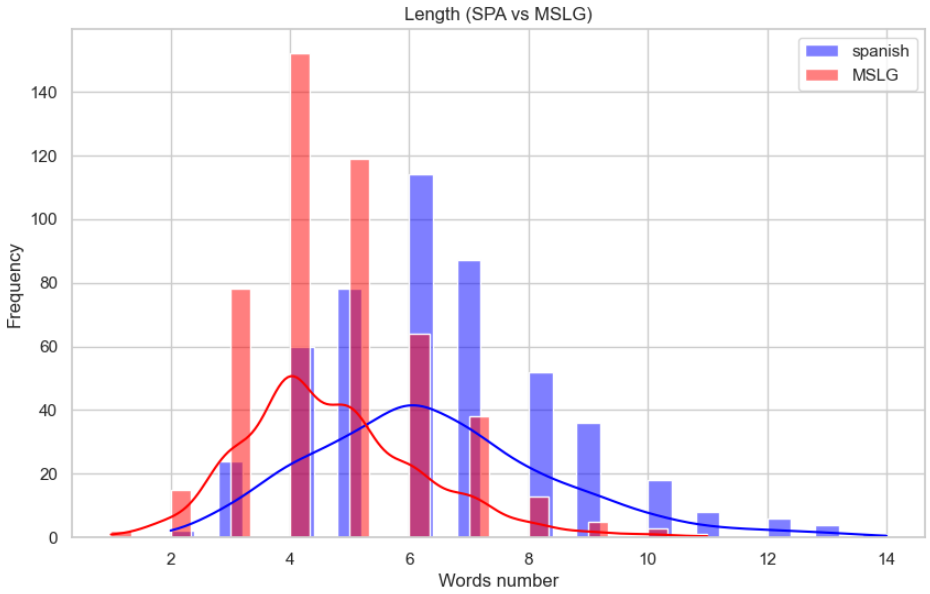

Figure 1: The histogram frequency of length words for both languages,at here, it can be seen that Spanish language has a greater length than MSLG

3. Proposed methodology

The proposed system is based on two transformer models [6]. The primary model is the T5 architecture [7], which is used for both training and evaluation, leveraging its encoder-decoder structure for sequence-to-sequence translation tasks.

Additionally, a BART model is incorporated as a refiner module that helps to complete the missing tokens in text generation. [8].

3.1. Ingest Data and Training T5 Model

The input data contains of parallel text pairs in Spanish and MSLG. As illustrated in the figure 2 the data is processed within the *Training T5 Model* block, specifically in the *Processing Raw Text* stage, which comprises 3 main components:

- Preprocessing
- Augmentation
- Normalization task

The preprocessing section, cleans the Spanish and MSLG text normalizing the whitespace, and in the specific case of MSLG, all tokens are converted to uppercase, and compound glosses with prefixes are transformer into their corresponding lexical forms. For example, *MAESTRO+MUJER* is converted into *MAESTRA*. The prefixes are stored with its meaning to be used furthermore in the generation and

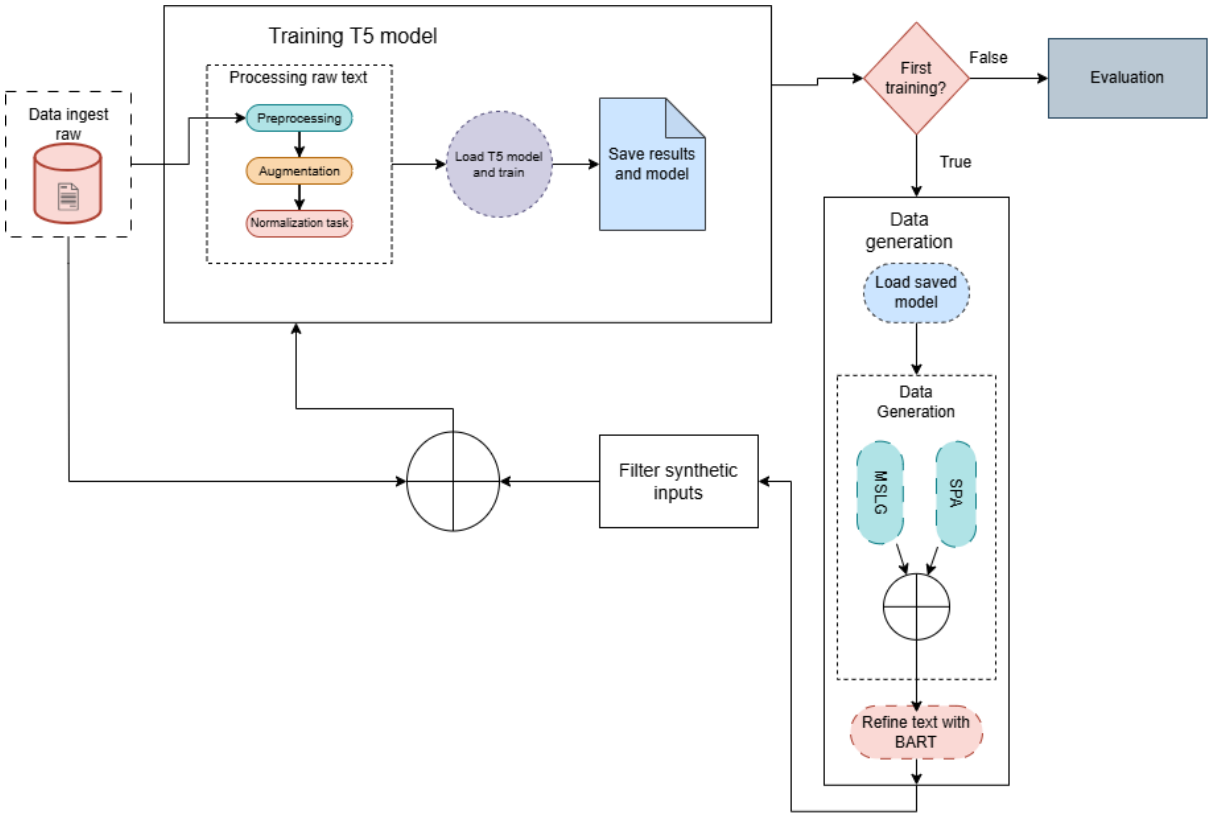

Figure 2: Complete system's pipeline

evaluation. The original dataset is split in 80% for train, 10% for validation and 10 % for testing. Each sample is assigned to one of two tasks: Spanish-to-MSLG translation or MSLG-to-Spanish. Due to the limited size of the dataset, additional samples are generated using Fast-text [9], with the pretrained Spanish embeddings *cc.es.300.bin* [10]. The final process is apply a normalization step by adding spaces around punctuation marks in Spanish text, ensuring consistency in tokenization. The processed dataset is then used to train the T5 model.

The model employed is the *T5-small*, which served as the backbone for the sequence to sequence translation task. Input text is tokenized with its respective T5 tokenizer based on subword units and, finally, to generate the translation, it requires a prompt indicating the type of task and the text desired to translate. Decoding is performed using beam search with a beam width of 4 and a maximum sequence length of 64 tokens. Model performance during training is evaluated using BLEU, METEOR, chrF, and COMET metrics, these metrics are calculated, as can be seen in the following equations

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1}$$

$$BP = \begin{cases} 1 & c > r \\ e^{1-r/c} & c \leq r \end{cases} \tag{2}$$

The equations 1 and 2 represent the standard form of BLEU, where the p_n is the n gram precision, the w_n are the weights, and BP is the brevity penalty that makes a penalization according to the length of the text. The letter c is the length from the predicted text and the letter r is the length from the original text [11]

$$METEOR = F_{mean} \cdot (1 - Penalty) \tag{3}$$

$$F_{mean} = \frac{10 \cdot P \cdot R}{R + 9P} \tag{4}$$

$$Penalty = 0.5 \left(\frac{c}{u_m} \right)^3 \tag{5}$$

In equations 3-5, the equations needed to calculate the METEOR metric; where the c is the chunk number text and u_m is the amount of matches in the translation. The P stands for precision and R for recall.[12]

$$chrF = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \tag{6}$$

The equation 6 represents another metric, which is the F score weighted; P is the precision of n grams, R is the recall of n_grams and β^2 is the weight recall [13] The COMET metric is a neural evaluation metric based in vector representations and similarity distances using token or sentence embeddings [14].

3.2. Generating New Inputs

After the first step of training, the model was saved. As the results obtained in the previous phase were high, one popular technique to improve the models in tasks where it just have a few amount of samples, is train and evaluate the model, furthermore, generate new data from the model trained, and concatenate this new knowledge with the original, this can be seen in the figure 2. There are several techniques to approach the issue of few samples [15], for example the Hyperparameter Tuning technique, Language Representation, Lexical Models usage are some of the typical techniques for Neural Machine Translation. [1] explores semi supervised techniques, Zero shot , back translation and transfer learning for multilingual tasks and parallel data, but [16] explores these approaches in monolingual tasks in non parallel[1] data approaches. This step uses a mix of these techniques, were the original text x is translated using the first version of T5 model,generating a $\hat{y}$ text. In other words, from a nonparallel data it applies the back-translation technique and generates a new dataset with pair-wised data, where it easily can be applied a semi-supervised model. These generated texts will be joined in a single dataset where a BART model will refine the text. This BART model, uses the named model *facebook bart-base*, and it uses its corresponding tokenizer.

3.3. Back-Loop Training step

The dataset with MSLG and the Spanish task was generated; the data is filtered. Due to the fact that T5 works with the help of a prompt, when it was initially trained, sometimes can input in the output the instruction, so the filter consist in removing that text. The cleaned text is joined to the original dataset and finally, the initial train loop talked in the section 3.1, will be applied to this new dataset where it will save the model and finally make the last evaluation, consisting in using the test dataset that has 245 examples for each task, and making the translation.

3.4. Package Versions and Model versions

To address the problem, it was used the programming language Python at the version 3.11. At also, the specific name model of t5 is `t5-small`, and the BART model used were `facebook/bart-base`. For the subgeneration task before the training with the help of fasttext, it used the model `cc.es.300.bin`. Other packages that were useful for the task were `bert-score`, `evaluate`, that were used for the initial metric calculations and `datasets (4.8.4)`, `transformers (5.6.2)`,these packages are useful for transformers and finally, the libraries `matplotlib (3.10.8)`, `numpy (2.4.4)`, `pandas (3.0.2)` that are useful for plots, numerical calculations and ingesting data

[1]Parallel data approach corresponds to have (x, y) pair-wised text, meanwhile non parallel represent unpair-wised text

Table 1
Evaluation metrics for the first step training model and the last model applying Self-learning techniques

Metric	Model1	Model2
Loss	2.5318	1.7022
BLEU	9.39	40.35
METEOR	0.2815	0.4830
chrF	36.5856	58.71
COMET	0.6091	0.7189

Table 2
Evaluation metrics for Out Of Vocabulary context with MSLG2SPA and SPA2MSLG tasks

Metric	MSLG2SPA task	SPA2MSLG task
BLEU	3.3651	3.1289
METEOR	0.1917	0.1863
chrF	29.1983	30.95
COMET	47.96	nan

4. Results

According to the table 1, the model 1 corresponds to the first model trained in the first stage,and it can be seen, that the results obtained in the metrics previously spoken, were lower than the results from the Model 2 that has the complete evaluation process. The BLEU score increased from 0.09 to 0.40, METEOR score increased from 0.2815 to 0.4830, chrF increased from 36.58 to 58.71 and COMET from 0.6091 to 0.7189. This shows that the last model improved the metrics and its performance with semi-supervised techniques as the usage of back-translation loop stage.

The results above correspond to the final stage of back-translation loop. However, when this last model is evaluated on an out of vocabulary benchmark, performance drops significantly, as shown in the table 2. This evaluation includes both MSLG-to-Spanish and Spanish-to-MSLG. Note the COMET metric was not calculated for SPA2MSLG task according to the guideline rules of the contest [3].

5. Discussion

The results obtained from the original train dataset confirm that the implementation of back-translation and semi supervised techniques, were useful to improve the performance across all evaluation metrics.

However, the results obtained from the benchmark test dataset, were completely different from the original train. This implies that the model with the semi supervised technique, learned semantic structure from the train loop, but for the out of vocabulary test, it reflects that the model can not generalize with unknown text or generate new compositions. In other words, although the proposed self learning framework significantly improves the performance on in-domain evaluation, the results in the last test reveal a limited generalization capability.

6. Conclusions and future work

This study demonstrates that the incorporation of semi-supervised learning techniques, particularly back-translation, leads to significant improvements in standard evaluation metrics. The proposed framework effectively enhances translation quality when the training and evaluation data share similar distributions.

However, an important insight derived from this work is that improvements in standard evaluation metrics do not necessarily translate into better generalization capabilities. While the back-translation strategy substantially improves performance within the training distribution, the results on the benchmark dataset reveal a critical limitation: the model exhibits poor performance when evaluated on

out-of-vocabulary inputs and under distribution shifts.

This behavior suggests that the gains obtained through semi-supervised learning are primarily driven by the reinforcement of existing patterns rather than the acquisition of more robust and transferable linguistic representations. As a result, the model becomes highly specialized in the training domain but lacks the flexibility required for more realistic or open-domain scenarios.

Several limitations of this work should be acknowledged. First, the model was evaluated under a limited set of conditions. Second, the reliance on synthetic data generated during the back-translation process may introduce bias and limit diversity, potentially contributing to the observed generalization gap. Third, the model architecture and training strategy were not extensively optimized. These findings highlight the need for future approaches to focus not only on improving evaluation metrics, but also on enhancing generalization capabilities. Future work will explore the use of more diverse datasets, domain adaptation strategies, and alternative architectures designed to better handle distributional variability. In particular, integrating parameter-efficient fine-tuning methods such as LoRA, as well as incorporating linguistic features such as Part-of-Speech (POS) information, could improve robustness. Additionally, leveraging pretrained models specialized in Spanish, such as BETO, may further enhance performance in domain-specific contexts.

Declaration on Generative AI

During the preparation of this work, the author use GPT based model and Grammarly in order to support language refinement and grammar correction

References

[1] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, R. Kaur, Neural machine translation for low-resource languages: A survey, 2021. URL: https://arxiv.org/abs/2106.15115. arXiv:2106.15115.

[2] A. Y. Rodríguez-González, D. Fajardo-Delgado, M. Á. Álvarez-Carmona, M. G. Sánchez-Cervantes, Á. Díaz-Pacheco, R. Aranda, S. Fajardo-Flores, C. H. Martínez-Rodríguez, Overview of mslg-spa at iberlef 2026: Bidirectional translation between mexican sign language glosses and spanish, Procesamiento del Lenguaje Natural 77 (2026).

[3] A. Bonet-Jover, J. Á. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS. org, 2026.

[4] Cámara de Diputados del H. Congreso de la Unión, Ley general para la inclusión de las personas con discapacidad, 2011. URL: https://www.diputados.gob.mx/LeyesBiblio/pdf/LGIPD.pdf, secretaría General, Servicios Parlamentarios, México.

[5] Secretaría de Educación Pública, Diccionario de lengua de señas mexicana, 2010. URL: http://www.utsoe.edu.mx/IGUALDAD/documentos/NMX-R-025-SCFI-2015%20en%20Igualdad%20Laboral%20y%20No%20Discriminaci%C3%B3n/Manuales%20de%20Lenguaje%20incluyente/Diccionario%20Lenguaje%20de%20Se%C3%B1as%20Mexicanas%201.pdf, méxico.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. URL: https://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL: https://arxiv.org/abs/1910.10683. arXiv:1910.10683.

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL: https://arxiv.org/abs/1910.13461. arXiv:1910.13461.

[9] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, 2016. URL: https://arxiv.org/abs/1607.01759. arXiv:1607.01759.

[10] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, 2017. URL: https://arxiv.org/abs/1712.09405. arXiv:1712.09405.

[11] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: https://aclanthology.org/P02-1040/. doi:10.3115/1073083.1073135.

[12] A. Lavie, K. Sagae, S. Jayaraman, The significance of recall in automatic metrics for mt evaluation, in: R. E. Frederking, K. B. Taylor (Eds.), Machine Translation: From Real Users to Research, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 134–143.

[13] M. Popović, chrF: character n-gram F-score for automatic MT evaluation, in: O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Pecina (Eds.), Proceedings of the Tenth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 392–395. URL: https://aclanthology.org/W15-3049/. doi:10.18653/v1/W15-3049.

[14] R. Rei, C. Stewart, A. C. Farinha, A. Lavie, COMET: A neural framework for MT evaluation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 2685–2702. URL: https://aclanthology.org/2020.emnlp-main.213/. doi:10.18653/v1/2020.emnlp-main.213.

[15] R. Sennrich, B. Zhang, Revisiting low-resource neural machine translation: A case study, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), 2019, p. 211–221. URL: http://www.acl2019.org/EN/index.xhtml, 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019 ; Conference date: 28-07-2019 Through 02-08-2019.

[16] B. Haddow, R. Bawden, A. V. Miceli Barone, J. Helcl, A. Birch, Survey of low-resource machine translation, Computational Linguistics 48 (2022) 673–732. URL: https://aclanthology.org/2022.cl-3.6/. doi:10.1162/coli_a_00446.

A. Online resources

Link to the code :https://github.com/JoseAdrianRodriguezGonzalez/MSLG-SPA-2026