Linguistically Enhanced *Text to Sign Gloss* Machine Translation

Santiago Egea Gómez[1], Luis Chiruzzo[2], Euan McGill[1], and Horacio Saggion[1]

[1] Universitat Pompeu Fabra, 08002 Barcelona, Spain
`{santiago.egea,euan.mcgill,horacio.saggion}@upf.edu`
[2] Universidad de la República, Uruguay, 11200 Montevideo, Uruguay
`luischir@fing.edu.uy`

Abstract. In spite of the recent advances in Machine Translation (MT) for spoken languages, translation between spoken and Sign Languages (SLs) or between Sign Languages remains a difficult problem. Here, we study how Neural Machine Translation (NMT) might overcome the communication barriers for the Deaf and Hard-of-Hearing (DHH) community. Namely, we approach the Text2Gloss translation task in which spoken text segments are translated to lexical sign representations. In this context, we leverage transformer-based models via (1) injecting linguistic features that can guide the learning process towards better translations; and (2) applying a Transfer Learning strategy to reuse the knowledge of a pre-trained model. To this aim, different aggregation strategies are compared and evaluated under Transfer Learning and random weight initialization conditions. The results of this research reveal that linguistic features can successfully contribute to achieve more accurate models; meanwhile, the Transfer Learning procedure applied conducted to substantial performance increases.

Keywords: Neural Transformers · Linguistic Features · Sign Gloss Machine Translation · Sign Language

1 Introduction

In the era of mass communication and wide uptake of digital technologies amongst the general public, there still exist barriers for many people where access to information is of concern, and this is particularly the case for the DHH community. Of the 466 million people worldwide with some kind of hearing loss [3], around 70 million communicate through SL[4] - the preferred mode of communication among the DHH people [21]. Recently, the European Commission adopted the Strategy for the rights of persons with disabilities[5], which indicates the need to provide SL interpretation to improve accessibility for the DHH community. There is a great opportunity for MT to bridge the gap between written/spoken languages and

[3] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss
[4] https://wfdeaf.org/our-work/
[5] https://ec.europa.eu/social/main.jsp?catId=738&langId=en&pubId=8376

SLs. But, in spite of the recent advances in MT for spoken languages, translation between spoken and SLs or between SLs (SLT) remains a challenge.

In terms of data and resource availability, SLs are considered 'extremely low resource' languages [13], which implies a salient issue for MT applied to SL. The latest approaches to MT are based on neural networks, particularly transformer models [22, 24]. Transformer-based systems are data-hungry and computationally expensive, so their viability for SLT is yet to be fully established.

Unlike spoken languages, a linear stream of information in the oral-auditory modality, SLs exist in the gestural-visual modality consisting of often parallel manual and non-manual cues [14]. This modality difference poses an important challenge in building corpora. Writing systems capturing the exact pattern and timing of signs are extant [8], however these systems are not widely used to annotate SL corpora nor are they widely known by signers [9]. Instead, SL glosses are preferred as an intermediate representation between SL video data and text in a MT paradigm (e.g. [5, 13, 23]). Glosses, and the Text2Gloss (T2G) process, are used as a tool to represent a given sign as a lexeme - usually in the ambient language of the geographical area where a SL is native[6]. In spite of criticism towards gloss annotation [23], one advantage of glosses in lexical signs is their suitability as a text representation to feed into machine learning models. These glosses can be represented within embedding vectors that are fed into neural models where mappings between the input and output texts can be established.

In our previous work [7], we showed how word dependencies can boost T2G translation. Here, we extend the experiments considering a range of linguistic features and different ways of injecting them into transformer models. We also show that Transfer Learning (TL) can be successfully applied to this particular MT task, spoken German to German Sign Language (DGS) translation.

2 Related Work

Translation between spoken and sign languages is not new and it has been investigated from several angles including example-based [12], rule-based [2], and statistical-based MT [18]. Transformer-based NMT models have been shown to be successful in producing translations for a wide range of language pairs with state-of-the-art accuracy [10], including low resource spoken languages [22]. Perhaps the most notable is mBART [24], a widely used pretrained transformer architecture for NMT. A key benefit of these powerful models is their ability to be fine-tuned to a downstream task in NLP such as MT.

SLT as a subset of MT is, however, a more challenging area. The task is inherently multimodal, and has severely limited resources. While E2E translation is possible from text to sign [6, 15], 'cascaded' systems involving intermediate steps appear to yield higher translation accuracy [5, 25]. Therefore it is important for the moment to focus on improving these intermediate stages. Cascaded building blocks for SLT may involve continuous SL recognition as a computer vision task,

[6] For example, glosses in written English for American Sign Language.

SL generation (e.g. [1]) from glosses into SL through 3D avatars, and Text2Gloss or Gloss2Text to facilitate translation through a text-to-text mapping.

A wide range of resources exist across multiple SLs such as parallel corpora, video corpora and repositories of signs produced in isolation. One problem is that for comparatively widely spoken languages, the size of these corpora are markedly smaller [9]. Co-occurring issues include domain specificity [5, 19] without examples of alternative semantic fields, little variation in signers both in diversity and positioning in 3D space [15] and a noise-free video background [25]. On top of cascading, breaking up SLT into a pipeline, there are further mitigation strategies to augment the training data available such as back-translation, previously avoided in SLT [25], or rule-based generation of glosses [13]. Alternatively, it is possible to augment the existing data with more information during training. This study follows the latter approach.

Sennrich and Haddow [20] introduced a 'Factored Transformer' model which inserts linguistic feature embeddings (lemmas, part-of-speech tags, lexical dependencies and morphological features) into the encoder of an attention-based NMT architecture. This schema was then used on a low-resource translation task English-Nepali [3] that improved performance on FLoRes[7] by 1.2 BLEU. Our recent work in Text2Gloss [7] explored the use of lexical dependencies in the model embeddings obtaining a peak improvement of 5.7 BLEU over a baseline.

Considering previous research, we hypothesize that linguistic features may boost model performances for T2G. We formulate three main research questions based on this prediction: (1) Which are the most informative features? (2) How do we inject them into transformer models? and (3) Can Transfer Learning provide performance increases when linguistic features are injected to the models? In order to shed light on these research questions, we propose a T2G system, analyse its performance, and discuss alternatives.

3 System Overview

The T2G translation system presented here is composed of three key components that Fig. 1 shows: (1) a *Text Processing* step to generate linguistic features to be injected to the model; (2) a *T2G Model* based on the mBart architecture [24]; and (3) a *Transfer Learning* process to get advantage of pretrained weights. We make the implementation of our system available in GitHub[8], along with an extended results analysis.

Considering the linguistic rules applied in the SL gloss production, we predict that linguistic features might play a relevant role in T2G translation task. Therefore, we use the available language resources to generate the linguistic features described in the Section 3.1. These features are aligned with the subword tokens generated by the mBART tokenizer as is depicted in Fig. 1. Unfortunately, there are not equivalent resources for DGS, an important restriction for us in exploring Gloss2Text translation.

[7] Facebook Low Resource benchmark

[8] https://github.com/LaSTUS-TALN-UPF/Linguistically-Enhanced-Text2Gloss-MT

As mBART architecture is difficult to manage due to its very large number of parameters, we have designed a strategy to take advantage of TL while using a much simpler neural architecture. Namely, we employ a transformer with 3 multi-attention layers with 4 heads for both the encoder and decoder, and the internal and output dimensions are set to 1024 and 512 respectively. Regarding the embeddings, we use 512-length vectors in two separate tables: One for sub-word tokens and another for linguistic features. The word and linguistic features are aggregated using different strategies which are also a matter of study of this research (Section 3.2).

In order to exploit the knowledge acquired by mBART during its multilingual pre-training, we filter and align the original mBART embedding table keeping only the tokens appearing in our training corpus. Additionally, the linguistic embeddings are initialized by randomly selecting vectors from mBART embeddings. As there are not control tokens to represent SLs in mBART, we reuse the Dutch language one to represent DGS; since German and Dutch are closely related languages. To fit the mBART weights into our architecture model, slicing is applied so that the vector elements are adapted to our neural architecture size.

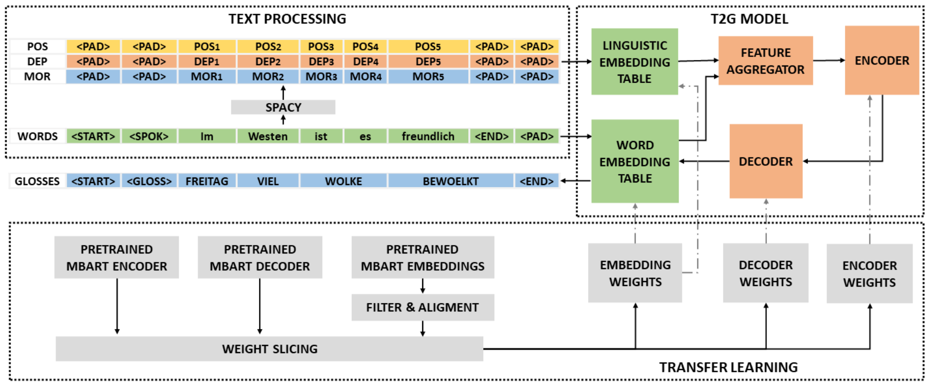

Fig. 1. An overview of the proposed system.

3.1 Linguistic Features

We labeled the input text with linguistic features using the `de_core_news_sm` model of the spaCy[9] library. This model is trained over the TIGER Korpus [4], a widely used German corpus partially annotated with POS, morphological information, and syntactic structure. The dependency labels used by spaCy are based on the TIGER Korpus format and differ from Universal Dependencies.

[9] https://spacy.io/

The linguistic features calculated for the input text are: Part-of-Speech (**POS**, 16 unique labels), dependency labels from the parse tree (**DEP**, 38 unique labels), and morphological information (**MOR**).

The tagger incorporates different types of morphological information for each word, including person and tense for verbs, and case and gender for nouns. We generate a tag that combines the different feature-value pairs into a single morphological label. For example, the noun *Westen* in the example from Fig. 1 would have the **MOR** label `Case_Dat-Gender_Masc-Number_Sing`. However, there is a large number of possible combinations of these features and their values: in the training corpus we found around 400 combinations, most of them used only a few times. To reduce the sparsity of this information, we used the following heuristic when creating the tag:

- If the combination of feature-value pairs of a word belongs to the top 60 most frequent combinations, we use the serialized combination described before (this covers around 66% of the corpus).
- Otherwise, we use the **POS** tag. This covers cases in which the morphological analyzer returns an empty value (around 32% of the words), so the labels can at least discriminate by POS in those cases.

As we used the mBART SentencePiece tokenizer [11], each word in the input might be split into one or more subword tokens, so the corresponding linguistic feature labels are associated to each of the tokens a word is split into.

3.2 Feature Aggregation Blocks

We aggregated the word information and the linguistic features in each experiment using different strategies. The following aggregation rules were applied to each linguistic feature separately (ablation study) and to all three jointly.

Reduce Sum & Product. These are very simple rules that enable combining the input features without adding new learnable parameters in the model.

Learnable Sum & Product. Learnable weights are incorporared into the ReduceSum&Prod operations. Namely, one learnable vector is multiplied by each input embedding to strengthen or weaken its contribution in the resulting vector.

Concatenation Mapping (ConcatMap). The embeddings are concatenated and fed to a dense network that maps the vector to a dimension of 512. The weights of the mapping network are also learned during training.

Convolutional Block (ConvBlock). As words and linguistic features could conceal complex relation patterns, a CNN network is employed with the aim of mining these complex relationships. Firstly, the input is reshaped to (32,16), and it is fed to the following layers: CNN_1(kernel = (8,8), channels = 4)-CNN_2(kernel = (8,8), channels = 4). Finally, the generated vectors are flattened and fed to a mapper network to adapt the dimensions to the encoder input shape.

Table 1. Data partitions Information

	Samples	**Words**		**Glosses**	
		Total	**Unique**	**Total**	**Unique**
Train	7096	99081	2887	55247	1085
Dev	519	6820	951	3748	393
Test	642	7816	1001	4264	411

4 Methods & Material

4.1 Phoenix Dataset

The corpus employed in our experiments is the *RWTH-PHOENIX-2014-T* [5], a very well-known SL corpus that is publicly available[10], which includes images and transcriptions in German text and DGS glosses. The data was extracted and annotated from weather forecasting news in a public TV station. And, in spite of its semantic limited domain, it comprises a rich vocabulary of 1117 different signs produced by nine different signers, being quite popular in SLT research. We focus on text and gloss data in this paper. The dataset is split into *train (≈ 86%)*, *development (≈ 6%)* and *test (≈ 8%)* partitions with the data distribution presented in Table 1. The partitions were created to ensure they contain a variety of syntactic structures.

4.2 Training & Evaluation Settings

Regarding the training settings, we train all models on the *train* data for 500 epochs using a learning rate of $5e^{-}6$. The learning objective is the *Cross Entropy loss function* without any regularization term at loss level. We apply generation on *development* partition using Beam Decoding with 5 beams. The best epoch model is selected and the scores are confirmed on *test* data. These steps are repeated 5 times for all models and the means and standard deviations are reported to avoid misleading observations due to randomness during training.

4.3 Performance Metrics

To evaluate our models, SacreBLEU [17] with word tokenization is assumed as main metric in model selection. SacreBLEU consists of a standardized version of traditional BLEU, which uses input tokenization to obtain more comparable results. This metric analyzes different N-grams against reference segments and aggregates them for a robust evaluation. Furthermore, we analyze other performance metrics during the test phase to have a wider understanding of how linguistic features contribute to the T2G task. The selected metrics are: SacreBLEU with character-level tokenization, which is used in our previous work [7]

[10] https://www-i6.informatik.rwth-aachen.de/ koller/RWTH-PHOENIX-2014-T/

and can be analyzed in a comparative manner. METEOR is also used, which evaluates the word alignments according to precision and recall using unigrams - giving recall higher importance to the metric computation.

5 Experimental Results

In this section we report and discuss the results obtained in our experiments involving the linguistic features (Section 3.1) and different feature aggregators (Section 3.2) under random initialization (RI) and Transfer Learning (TL) settings. First, we compare the aggregators under RI; then, TL is explored for the best aggregation rules; finally, we study individually the advantage of each linguistic feature. Afterwards the performances are analyzed on the *test* partition.

5.1 Comparison amongst Aggregation Blocks

Here, we compare the different aggregation rules including a model (*OnlyWords*) as baseline, which only uses the word embedding table. Fig. 2 presents the SacreBLEU scores obtained on the *development* partition during the whole training. As can be observed, the best aggregation strategy is *ConvBlock*, which out-

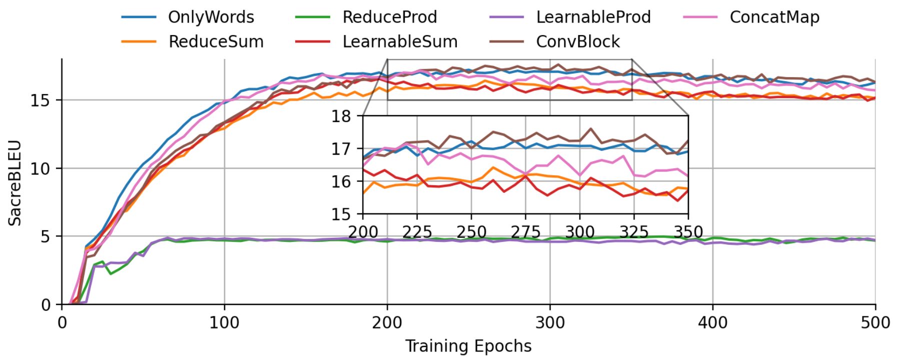

Fig. 2. SacreBLEU curves for the different aggregation blocks with random initialization.

performs *OnlyWords* at certain epochs. The highest SacreBLEU produced by *ConvBlock* is 17.59 after 305 epochs, overcoming the best *OnlyWords* (17.21 at 270 epochs). In the case of *ConcatMap*, the highest SacreBLEU happens at 220 epochs obtaining a score very similar to *OnlyWords* (17.13), but slightly lower. The rest of aggregation rule models perform worse compared to *OnlyWords*, with *LearnableProd* and *ReduceProd* as clearly the poorest aggregators. The reason why the aggregation rules using prod obtained the worst scores might be due to the scale in the produced embedding vectors.

As *ConvBlock*, *OnlyWords* and *ConcatMap* are the three best models, we compare them using TL and RI in the following section.

5.2 Transfer Learning vs Random Initialization

Fig. 3 presents the SacreBLEU curves obtained for the three best models (selected from the previous experiment) when TL and RI is employed. The effect

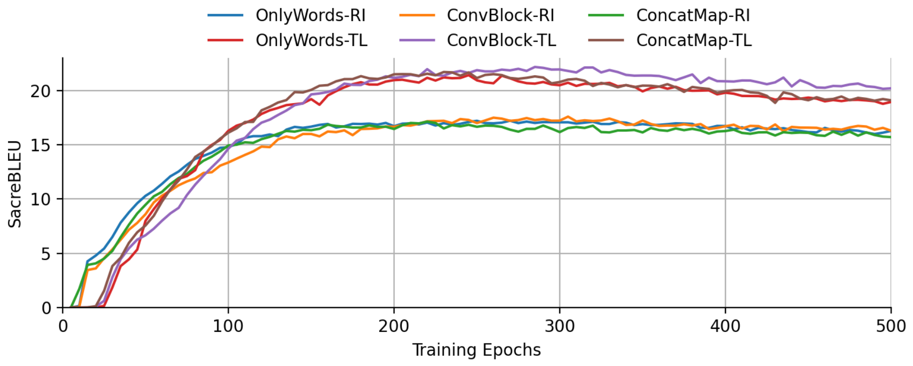

Fig. 3. SacreBLEU curves for the three best models with and without Transfer Learning.

of TL is evident for all models, and the improvements respecting RI reach up to 5 sacreBLEU in certain cases. Again, the best model is produced by *ConvBlock* aggregation rule when TL is applied, achieving a SacreBLEU of 22.17 at 285 training epochs. The benefits of TL are also marked for *ConcatMap* and *OnlyWords*, obtaining SacreBLEUs of 21.70 and 21.41 respectively. These performance increases confirm that the TL strategy applied to our models allows them to take advantage of the pre-learned knowledge of mBART, but using a more manageable architecture. Also interestingly, *ConvBlock-TL* exhibits better performances than other aggregation strategies, which reveals that the CNNs are extracting patterns that enrich the embeddings input into the encoder.

5.3 Ablation Study: Comparison amongst Linguistic Features

In this section, we assess the contributions of each linguistic feature for the translation task. To this end, we compare the SacreBLEU produced by different aggregation rules when each linguistic feature is individually injected. This analysis is performed anew with focus on the best models found during the previous experiments. Thus, we compare *ConcatMap-TL* and *ConvBlock-TL* using all (denoted by ALL) and individual features; and, additionally, we include the *OnlyWords-TL* as a baseline. The metric curves for these models are presented in Fig. 4. As it can be observed, the best models are generated using *ConvBlock*

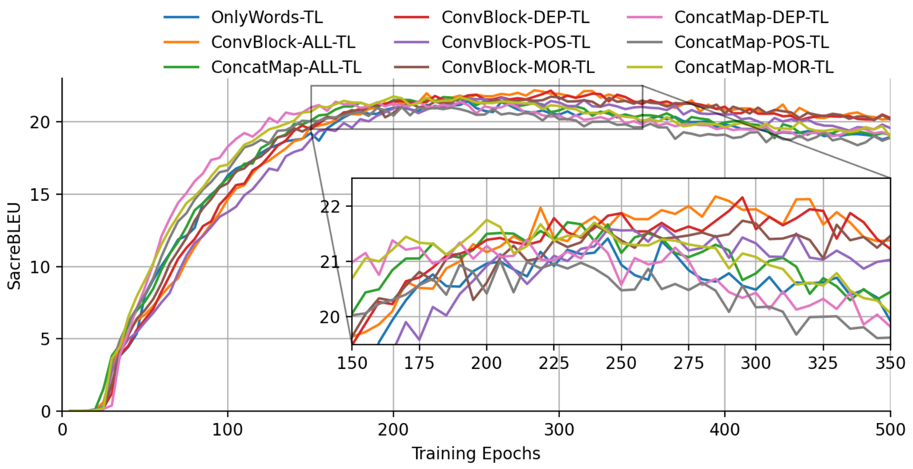

Fig. 4. SacreBLEU curves for the best models using all linguistic features and each one individually

with *DEP* and *ALL* features. Also, the differences between these two models are not substantial. Therefore, *ConvBlock* extracts rich embedding vectors using only *DEP* and without needing to include *MOR* and *POS*.

Analyzing each feature individually, we find that the models using *POS* obtained lower SacreBLEU than the rest of the linguistic features. Moreover, the *ConvBlock-POS-TL* curve is below the *OnlyWords-TL* at many training points. Regarding *MOR* models, their SacreBLEU scores are very close to *DEP* and *ALL* models, but without any superior performances compared to them.

Finally, it is also important to highlight the differences between aggregation blocks. The behavior of *ConvBlock* differs from the *ConcatMap* in the fact that the former starts learning slower than the latter (which is clearly observable between $50 - 200$ training epochs). On the contrary *ConvBlock* scores overcome the *ConcatMap* ones, showing more stability after 250 epochs.

5.4 Results on *test* data

To conclude our study, we analyze the scores over the *test* partition of the models that had best SacreBLEU on *development*. In this case, we include additional performance metrics that might reveal relevant behaviors about the models analyzed. From Table 2, we can observe that the improvements achieved by the TL strategy are also noticeable on the *test* data for all the aggregation rules and all performance metrics. Comparing *ConvBlock-ALL-TL* & *RI*, we find that SacreBLEU-Char increases up to more than 5 points in the case of *ConvBlock* and METEOR improves by around .052. These improvements are also evident for *ConcatMap* and *OnlyWords* models.

Contrary to the findings on *development*, the effect of using *ALL* features and each feature individually is not so notable in the case of *ConvBlock*. Using

Table 2. Performance metrics computed on *test*. We report average metric (and standard deviation) for 5 experiment iterations. SacreBLEU-Word & Char denote the different tokenization methods for this metric.

	SacreBLEU-Word↑	**SacreBLEU-Char↑**	**METEOR↑**
OnlyWords-RI	16.61 (.664)	52.70 (.664)	.409 (.005)
OnlyWords-TL	19.89 (.441)	56.01 (.783)	.454 (.005)
ConvBlock-ALL-RI	16.72 (.608)	52.63 (.805)	.406 (.005)
ConvBlock-ALL-TL	20.24 (.976)	57.30 (.365)	.458 (.005)
ConvBlock-DEP-TL	20.11 (.310)	<u>57.45</u> (.406)	.460 (.004)
ConvBlock-POS-TL	19.94 (.526)	56.75 (.390)	.456 (.004)
ConvBlock-MOR-TL	20.36 (.759)	<u>57.45</u> (.565)	<u>.461</u> (.005)
ConcatMap-ALL-RI	16.77 (.298)	51.79 (.521)	.408 (.003)
ConcatMap-ALL-TL	<u>20.57</u> (.737)	56.53 (.375)	.458 (.005)
ConcatMap-DEP-TL	19.56 (.760)	56.49 (.865)	.452 (.005)
ConcatMap-POS-TL	19.69 (.885)	56.82 (.746)	.453 (.006)
ConcatMap-MOR-TL	19.30 (.392)	56.20 (.702)	.450 (.005)

ALL features with *ConcatMap* result in increases of up to 1 point in SacreBLEU-Word. Meanwhile, similar results are seen in METEOR and SacreBLEU-Char.

Globally, we can observe that *ConvBlock* produces better performance than other aggregation rules for all settings and metrics explored, with the exception of *ConcatMap-ALL-TL* in terms of SacreBLEU-Word. This result may be caused by the simple architecture tuning explored in this research. We posit that it might be possible to enhance the pattern mining with CNNs including more layers and regularization techniques. Finally, considering our previous research [7], we can observe a substantial improvement of around 4 points in SacreBLEU-Char and 0.6 in METEOR when CNNs and TL is applied.

6 Conclusions & Future Work

In this paper, we study the potential of injecting linguistic features into neural transformers for a T2G translation task. The experiments presented involve several types of linguistic features which are aggregated to the traditional subword embeddings according to different aggregation strategies. These strategies comprise of simple rules (such as ReduceSum & Prod) and more sophisticated aggregators able to mix the features while extracting hidden patterns (CNNs, Learnable Sum & Prod, and so on). Furthermore, we show that TL can robustly improve the performance of T2G models via applying a simple, but effective, filtering and slicing procedure.

According to our results, using CNNs or concatenating features produces the best results. Regarding the features to include in the models, we find interesting improvements when using all available linguistic features on *development* data, but this is not so clear for the *test* data. Finally, the most remarkable performance improvements are produced when TL is applied according to the method described in this paper, resulting in improvements on all metrics. For the sake of research reproducibility, we make our implementations available in GitHub[11].

The experimental results reported here leave room for interesting future research that may be materialized in the following research lines. (1) Approaching the translation task in the other direction (Gloss2Text), which requires the production of linguistic resources to annotate SLs, (2) Extending the experiments to other SLs and SL corpora, which could involve multilingual settings, and (3) Integrating multimodal features, such as manual and/or non-manual information, and visual features. For achieving (1), it will be necessary to create a tagger and a dependency parser for DSG, which would imply annotating many resources. This has been tried for other languages in the past (e.g. for Swedish Sign Language [16]), but so far existing corpora annotated in this way is very scarce, making this a very challenging problem.

Acknowledgements This work has been conducted within the SignON project. SignON is a Horizon 2020 project, funded under the Horizon 2020 program ICT-57-2020 - "An empowering, inclusive, Next Generation Internet" with Grant Agreement number 101017255.

References

1. Almeida, I., Coheur, L., Candeias, S.: From European Portuguese to Portuguese Sign Language. In: 6th WS on SLPAT. pp. 140–143. ACL, Dresden, Germany (Sep 2015). https://doi.org/10.18653/v1/W15-5124
2. Almeida, I., Coheur, L., Candeias, S.: Coupling natural language processing and animation synthesis in portuguese sign language translation. In: 4th WS on VL. pp. 94–103 (01 2015)
3. Armengol Estapé, J., Ruiz Costa-Jussà, M.: Semantic and syntactic information for neural machine translation: Injecting features to the transformer. Machine Translation **35:3**, 3–17 (2021). https://doi.org/10.1007/s10590-021-09264-2
4. Brants, S.e.a.: TIGER: Linguistic interpretation of a german corpus. J. of Lan. and Comp. **2**, 597–620 (12 2004). https://doi.org/10.1007/s11168-004-7431-3
5. Camgoz, N., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: CVPR 2018. pp. 7784–7793 (03 2018). https://doi.org/10.1109/CVPR.2018.00812
6. Camgoz, N.C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. In: CVPR 2020. pp. 10020–10030 (2020). https://doi.org/10.1109/CVPR42600.2020.01004
7. Egea Gómez, S., McGill, E., Saggion, H.: Syntax-aware transformers for neural machine translation: The case of text to sign gloss translation. In: 14th WS. on BUCC. pp. 18–27. INCOMA Ltd., Online (Sep 2021), https://aclanthology.org/2021.bucc-1.4

[11] https://github.com/LaSTUS-TALN-UPF/Linguistically-Enhanced-Text2Gloss-MT

8. Hanke, T.: Hamnosys—representing sign language data in language resources and language processing contexts. In: LREC 2004, WS on RPSLs. pp. 1–6. Paris, France (05 2004)
9. Jantunen, T., Rousi, R., Raino, P., Turunen, M., Valipoor, M., García, N.: Is There Any Hope for Developing Automated Translation Technology for Sign Languages?, pp. 61–73 (03 2021). https://doi.org/10.31885/9789515150257.7
10. Jurafsky, D., Martin, J.H.: Speech and language processing (3rd edition draft). https://web.stanford.edu/ jurafsky/slp3/ (2021)
11. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: EMNLP 2018. pp. 66–71. ACL, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/D18-2012
12. Morrissey, S., Way, A.: An example-based approach to translating sign language. In: 2nd WS on Example-based MT (Sept 2005)
13. Moryossef, A., Yin, K., Neubig, G., Goldberg, Y.: Data augmentation for sign language gloss translation (2021), https://arxiv.org/abs/2105.07476
14. Mukushev, M., Sabyrov, A., Imashev, A., Koishybay, K., Kimmelman, V., Sandygulova, A.: Evaluation of manual and non-manual components for sign language recognition. In: LREC 2020. pp. 6073–6078. ELRA, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.lrec-1.745
15. Nunnari, F., España-Bonet, C., Avramidis, E.: A data augmentation approach for sign-language-to-text translation in-the-wild. In: LDK 2021, Zaragoza, Spain. OpenAccess Series in Informatics (OASIcs), vol. 93. Dagstuhl publishing (9 2021)
16. Östling, R., Börstell, C., Gärdenfors, M., Wirén, M.: Universal dependencies for swedish sign language. In: Proceedings of the 21st Nordic conference on computational linguistics. pp. 303–308 (2017)
17. Post, M.: A call for clarity in reporting BLEU scores. In: 3rd Conf. on MT. pp. 186–191. ACL, Belgium, Brussels (Oct 2018). https://doi.org/10.18653/v1/W18-6319
18. San-Segundo, R.e.a.: Design, development and field evaluation of a spanish into sign language translation system. Pattern Anal. Appl. **15**(2), 203–224 (may 2012)
19. San-Segundo, R., López, V., Martín, R., Sánchez, D., García, A.: Language resources for spanish - spanish sign language (lse) translation. In: 4th WS on RPSLs. pp. 208–211 (2010), http://www-gth.die.upm.es/research/documentation/AG-096Lan-10.pdf
20. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. In: 1st Conf. on MT. pp. 83–91. ACL, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/W16-2209
21. Stoll, S., Camgöz, N.C., Hadfield, S., Bowden, R.: Text2sign: Towards sign language production using neural machine translation and generative adversarial networks. IJCV **128**(4), 891–908 (2020). https://doi.org/10.1007/s11263-019-01281-2
22. Xue, Linting et al.: mT5: A massively multilingual pre-trained text-to-text transformer. In: NAACL 2021. pp. 483–498. ACL, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.41
23. Yin, K., Read, J.: Better sign language translation with STMC-transformer. In: COLING 2020. pp. 5975–5989. ICCL, Online (Dec 2020). https://doi.org/10.18653/v1/2020.coling-main.525
24. Yinhan Liu, et al.: Multilingual denoising pre-training for neural machine translation. CoRR pp. 1–17 (2020), https://arxiv.org/abs/2001.08210
25. Zhang, X., Duh, K.: Approaching sign language gloss translation as a low-resource machine translation task. In: AT4SSL 2021. pp. 60–70. AMTA, Online (Aug 2021), https://aclanthology.org/2021.mtsummit-at4ssl.7