Sign Language Translation

Erkan Turan - Aissa Abdelaziz

15-01-2024

Abstract

This project centers around Sign Language Translation, aiming to convert continuous sign language videos into spoken language sentences. The task is challenging due to its inherent multi-modal nature, the complexity and variability of sign language, the need for sophisticated transformer-encoder architectures requiring expensive training, and the typically limited training data. The project involves: 1) training model with the provided code to reproduce the reported results, 2) investigating the model's performance when transitioning from I3D to SWIN extracted video features, 3) implementing a text data-augmentation strategy with a paraphrasing module, and 4) introducing and qualitatively studying new evaluation metrics (ROUGE and BERT-score) before incorporating them into the codebase.

1. Introduction

Sign Language Translation is a challenging multi-modal problem with clear and well-motivated applications which could bring important advances in the context accessibility and communication. One aspect which make this problem difficult is the limited availability of high-quality annotated data. Prior work measured SLT performance on restricted tasks by training models on the Phoenix-2014T [5] dataset which represents a mere 9 hours of weather forecast footage. Recently Larre's et al. [7] reportedly introduced the first baseline model, consisting of a standard transformer encoder-decoder architecture trained on the more involved How2Sign dataset [2] which represents 80 hours of footage spread over 10 different topics. They achieved a BLUE-score of 8.03. This preliminary work offered a lot of room for improvement, indeed no text-data augmentation strategies where employed, this could help alleviate the data-scarcity and offer better generalization power by encapsulating language variability by including paraphrase at training time. Additionally, the work used features-extracted from an inflated 3D convolutional neural network architecture [1] pre-trained on ImageNet and then fine-tuned on Kinetics-300 and How2Sign. Recently the vision community is witnessing a modeling shift from CNNs to Transformers, where pure Transformer architectures have attained top accuracy on the major video recognition benchmarks including on the Kinetics-300 dataset. Replacing features extracted from the I3D backbone for SWIN features [4] seems like a promising direction for enhancing performance. Lastly, the paper put forward the importance of carefully choosing a metric as certain raw numbers can be quite misleading in the correct assessment of model performance, integrating different measures therefore seems like a sensible option. Before exploring these various strategies we first get accustomed with their provided code to reproduce their declared result.

2. Reproducing of results and SWIN features

2.1. Reproduction of the results

The first step in this work was to reproduce the declared result of 8.03 BLUE-score. The code was provided but the codebase was quite messy, indeed multiple discrepancies existed between the source code and the documentation making this process slightly more tedious than expected. After thoroughly dissecting the code base, rewiring certains pathways and making the adequate downgrades on approriate packages we managed to train their proposed encoder-transformer architecture and got the BLUE-score reported in Tab. 1. Figure 1. shows the learning curve obtained from this initial step. We don't get exactly the number reported in the original paper, after discussions with other groups it is possible that the discrepancy comes from the use of the included generate.py script.

	BLEU	Reduced-BLEU
Reported	8.03	2.21
Obtained	7.79	2.07

Table 1. The reported and achieved BLEU scores after training. Score on test set

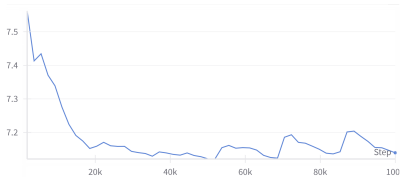
2.2. Replacing I3D features with SWIN features

In the original work the video features are extracted from an inflated 3D convolutional neural network backbone [1] that was pre-trained on ImageNet and then finetuned on the Kinetics-300 and the How2Sign dataset [2]. Recently there has been a paradigm shift in the vision community from the traditional convolutional to attention-based neural architectures which manage to attain top accuracy on major benchmarks. Recently an adapted Video SWIN transformer [4] has been introduced for the task of action recognition and has achieved new state of the art. In this context it is sensible to think that replacing the features extracted from the I3D backbone with features coming from the adapted Video SWIN transformer may yield some performance gain on the SLT task, we decided to investigate this idea. SWIN features were provided to us, to ease their integration in the pipeline we had to perform two following two steps: 1) adapt the directory structure of the provided feature file to make it compatible with the original I3D feature file structure. 2) reformat the SWIN features. The first step being straightfoward we'll only detail the second point. The provided I3D and SWIN video features were different in nature. Indeed the I3D video feature consisted of a video feature per sentence, the start of the sentences were located in time via timestamps t_i. On the other hand the SWIN features consisted of feature per video taken by sliding a window of 16 frames with a stride of 2. In order to seemlessly integrate the SWIN features in the provided codebase we had to reformat the SWIN features in accordance with the I3D features, to do so we had to convert the I3D timestamps to SWIN feature index i using the following formula:

$$i = \lfloor \frac{fps \times t_i - c}{s} \rfloor \quad (1)$$

Where $\lfloor\ \rfloor$ denotes the whole part, fps denotes the video frame-rate, c the window center (7 here) and s the stride. This formula in hand we were able to reshape the SWIN features into the I3D form expected by the codebase. We integrated these features into the training pipeline and made sure to modify the size of the linear layer taking these features as input from 1024 to 768. Having performed these changes, we then trained the network. Upon visual inspection of the learning curve provided in Fig 1. it appears that the model is learning which provides a good sanity check on our implementation. After the 108 training epochs we notice on Tab. 2 a big decrease in performance compared to the original pipleline using I3D features, indeed BLEU score goes from 7.79 to 1.47. This loss in performance could possibly be explained by the lack of fine-tuning of the Video SWIN Transformer to the How2Sign dataset compared to the I3D backbone. It could have been interesting to measure the impact of such a fine-tuning but as this step required downloading the How2Sign dataset and training a sophisticated transformer architecture which is computationnaly expensive this was left unexplored.

Figure 1. Loss during training phase with SWIN features



	BLEU	Reduced-BLEU
SWIN	1.47	0.03
I3D	7.79	2.07

Table 2. The reported score BLEU-score with I3D features and the achieved BLEU-score after training with the SWIN features. Score on test set

3. Text data-augmentation

In the original work no text data-augmentation strategies were employed. As mentioned earlier, data scarcity coupled with sign language variability make it difficult for a model to generalize well in the context of sign language translation and may make it prone to overfitting. The authors actually observed this behavior and relied on heavy regularization techniques such as employing dropout, weight decay and label smoothing. In this section we investigate the impacts of integrating a paraphrasing module in the training routine. Four strategies where investigated: 1) naive thesaurus-based substitution, 2) word embeddings substitution 3) utilizing a masked language model and finally 4) back-translation. We briefly describe this strategies.

3.1. Naive Thesaurus-based Substitution

The idea of Naive Thesaurus-based substitution consists in randomly taking a word in the sentence and replacing it with a synonym using a Thesaurus. For implementation, we used the WordNet database and the the TextBlob API which provides a programmatic access to WordNet. This method has the advantage of being very fast to execute but has the disadvantage of not taking into account sentence context which sometime yields to poor rephrasing.

3.2. Word Embeddings Substitution

In this approach, we take pre-trained word embeddings such as Word2Vec, GloVe, FastText, Sent2Vec, we chose a random word in the sentence and use the nearest neighbor words in the embedding space as the replacement. For implementation we used the *gensim* API, a Glove word embedder which was trained on a Twitter feed. This method is also very ef-

ficient but suffers the same problems from Naive Thesaurus-based Substitution.

3.3. Masked Language Model

This method relies on employing transformers that have been trained on a large amount of text using a pretext task called "Masked Language Modeling" where the model has to predict masked words based on the context. We use this to paraphrase text by randomly masking a word in a given sentence and letting the model predict the missing token. Because long-range dependencies are naturally incorporated in these architectures via the attention-based mechanism this technique has the advantage of yielding coherent paraphrases. We used the Hugging Face's transformer library to test out this method. Unfortunatly this method was quite slow in our testing.

3.4. Back-translation

In this approach, we leverage machine translation to paraphrase a text while retaining the meaning. The idea consists of taking a sentence, translating into a language and then translating back to the orignal language. In our testing we performed backtranslation with Italian, French and German and performed all possible translation combinations between these language such as English - French – English or English - French - German - Italian - English, etc... Via this strategy we managed to generate on average 10 high quality paraphrase from each sentences. This route was very promising for training but unfortunately the generation step was quite slow and considering the size of the training dataset of 30000 sentence entries this method was too involved. In the end we utilized Google Sheets and there built-in Google-Translate function. We performed back-translations between English-French and English-Italian, by discarding any possible duplicates we still achieved a data increase by a factor of 2.43.

3.5. Training and Results

All methods have been qualitatively assessed with sentences taken from the training data. Paraphrasing examples are reported in Tab. 4 and Tab. 5. This assessment motivated the usage of back-translation as a paraphrasing module as the three other methods can, on certain occasion, alter the meaning of the sentence in an uncontrolled way. After training on this augmented dataset we obtain the results reported in Tab. 3. We measure a decrease in performance, this might be corrected by modifying hyperparameters which play a crucial role, as reported in the original paper. Decreasing regularization could have been something to look into but was left unexplored in this work.

	BLEU	Reduced-BLEU
Baseline	7.79	2.07
Data-Augmentation	7.28	1.7

Table 3. The reported and achieved BLEU scores after training with data-augmentation strategy. Score on test set

4. Metrics

In the paper they put forward the importance of choosing an appropriate metric, they opt for BLEU and reduced-BLEU [6]. This last metric allows to alleviate the bias towards correctly translating very frequent words that have limited importance in the significance of the sentence. In this section we investigate other measures, namely the BERT-score [8] and ROUGE [3]. After a qualitative assessment of their specificities we integrated them in the evaluation script to give us an indication of performance relative to these scores, we also coupled them with the paraphrasing module to evaluate the quality of the translation against the ground-truth and synthetic paraphrases.

4.1. ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the quality of summaries generated by automatic summarization and machine translation systems. The metrics are designed to measure the overlap between the generated summary and one or more reference summaries. They are calculated as follows:

$$RECALL = \frac{\text{\# of overlapping n-grams}}{\text{\# of overlapping n-grams in the reference}} \quad (2)$$

$$PRECISION = \frac{\text{\# of overlapping n-grams}}{\text{\# of overlapping n-grams in the candidate}} \quad (3)$$

$$F1 = \frac{2\text{PRECISION} \times \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \quad (4)$$

It has the following advantages of providing an objective and flexible measure, but due to its reliance on overlapping it is quite sensitive to synomym replacement and paraphrasing. This can be problematic in the context of translation where general meaning supersedes exact presence of n-grams when evaluating performance.

4.2. BERT-score

Calculating the BERT-score relies on the following steps: 1) represent both the reference and candidate sentences using contextual embeddings 2) compute pairwise cosine similarity between each token in the reference sentence and each token in the candidate sentence 3) each token in the reference

sentence is matched to the most similar token in the candidate sentence, and vice versa, to calculate recall and precision which are then combined in the F1-score. The main advantage of this metric is the use of contextualized embedding which helps in capturing nuances and contextual information. A disadvantage is the lack of interpretability confered by the black-box nature of the BERT model.

4.3. Illustrative example

We used Hugging Face's implementation of the ROUGE and BERT scores. We qualitatively assessed their qualities on a few ground-truth sentences with the associated model predictions obtained after training. We report in Tab. 7 the scores given for a correct translation, a satisfactory translation and a poor translation. Upon manual inspection, it seems that in clear situations both metrics are fairly aligned in terms representing bad or good translation. But as seen in the table some discrepancies may arise in more ambiguous cases. It seems that the BERT-score might be more reliable in such an SLT context where predictions can have the same meaning without necessarily presenting the n-grams as the reference. It could have been interesting to keep the best model with respect to this measure and to analyze the performance of the obtained model.

5. Conclusion

In this study of Sign Language Translation, we successfully navigated existing code, trained a standard transformer encoder-decoder in a multi-modal setting. We explored the impact of using alternate video features which put to light the importance of fine-tuning in computer vision tasks. Despite no observed performance enhancement with text data-augmentation, this observation emphasized the need for combining quality data with a careful tuning of hyperparameters. The work underscores the importance of selecting evaluation metrics judiciously. Possible extensions include further fine-tuning the Video SWIN Transformer on the How2Sign dataset, optimizing hyperparameters for training on the augmented dataset , and possibly keeping best model checkpoint with respect to other metrics to evaluate differences in translation performance.

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 1, 2

[2] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[3] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*, pages 74–81, 2004. 3

[4] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 1, 2

[5] Oscar Koller Hermann Ney Richard Bowden Necati Cihan Camgöz, Simon Hadfield. Rwth-phoenix-weather 2014 t: Parallel corpus of sign language video, gloss and translation. In *Neural Sign Language Translation, IEEE Conf. on Computer Vision and Pattern Recognition,*, pages 5625–5635, 2018. 1

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311–318, USA, 2002. Association for Computational Linguistics. 3

[7] Laia Tarrés, Gerard I. Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i Nieto. Sign language translation from instructional videos. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5625–5635, 2023. 1

[8] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. 3

Method	Reference	Good Paraphrase
Thesaurus-based	But **fortunately** it's very fixable	But **luckily** it's very fixable
Word-embedding	I mean this is **nice**	I mean this is **good**
Masked-Language-Model	In **this** next clip we're going to be going over a hook kick. Ok. A hook is a very powerful kick	In **the** next clip we're going to be going over a hook kick. Ok. A hook is a very powerful kick
Back-Translation	It also increases the blood **circulation** to your brain which **nourishes** your papilla.	It also increases the blood **flow** to the brain that **feeds** the papilla.

Table 4. Examples of good paraphrasing yielded by the different strategies

Method	Reference	Bad Paraphrase
Thesaurus-based	My name is Dr. **Art** Bowler.	My name is Dr. **fine art** Bowler
Word-embedding	The kids will love **you** for it	The kids will you **you** for it
Masked-Language-Model	So we're **hitting** the four corners of the room	So we're **occupying** the four corners of the room

Table 5. Examples of bad paraphrasing yielded by the different strategies. Back-translation wasn't included as no such anomalies was observed

Method	Execution Time per Paraphrase in seconds
Thesaurus-based	0.002s
Word-embedding	0.01s
Masked-Language-Model	0.04s
Back-Translation	0.29s

Table 6. Execution time in seconds per paraphrase for the various methods

Reference	Predicted	ROUGE-L score	BERT-score
one. two. three	one. two. three	1	1
again i want to make faces.	again, we're going to make the face	0.43	0.91
keep your eyes positioned up	you're going to push it down.	0	_

Table 7. Examples of ROUGE-L and BERT scores for various translation quality. From top to bottom we have: perfect translation, satisfactory translation, bad translation