Machine Translation from Spoken Language to Sign Language using Pre-trained Language Model as Encoder

Taro Miyazaki, Yusuke Morita, Masanori Sano
NHK Science and Technology Research Laboratories
1-10-11 Kinuta, Setagaya-ku, Tokyo, Japan
{miyazaki.t-jw, morita.y-gm, sano.m-fo}@nhk.or.jp

Abstract
Sign language is the first language for those who were born deaf or lost their hearing in early childhood. Such individuals understand sign language better than transcribed spoken language because sign languages differ from spoken languages in not only the modalities to express meanings but also in grammar and vocabulary. Therefore, they require services provided with sign language, but there are only a few services provided. For example, less than 0.5% of airtime of TV programs have sign language services in Japan (Ministry of Internal Affairs and Communications, 2019). There are many efforts to develop systems to provide more services in sign language through computer graphics (CG)-based animation (Kipp et al., 2011; Romeo et al., 2014; Uchida et al., 2018; Azuma et al., 2018). These systems are designed for practical domain-specific services. Therefore, they apply rule-based translation methods. Typical rule-based translation methods can translate with high quality for the target domain, but the coverage for the input tends to be narrow.

To provide sign language services that can be used for flexible, open-domain target contents, non-rule-based machine translation is necessary. Machine translations generally require large-scale training corpora (Koehn and Knowles, 2017; Lample et al., 2018). However, there are only small corpora for sign languages; one reason is that sign languages do not have writing systems. To overcome this data-shortage scenario, we use a pre-trained language model of spoken language for the machine translation model. Our method is based on Transformer (Vaswani et al., 2017), and we use BERT (Devlin et al., 2019) as the initial model of the encoder. The encoder embeds the input transcribed spoken language, then the embedded vectors are fed to the decoder, which is also based on Transformer, then the input sentences are translated into sign language glosses. Evaluation results indicate that our method outperformed baseline methods, including phrase-based statistical machine translation (PBSMT)-based method, using only 130,000 sentence pairs of training data. We also show that one of the reasons of translation error is from Pronoun, which is a special feature used in sign language. We also conducted trials to improve the translation quality for Pointing. The results are somewhat disappointing, so we believe that there is still room for improving translation quality, especially for Pointing.

Keywords: Japanese Sign Language, Machine Translation, BERT, Pointing

1. Introduction
Sign language is the first language for those who were born deaf or lost their hearing in early childhood. Such individuals understand sign language better than transcribed spoken language because sign languages differ from spoken languages in not only the modalities to express meanings but also in grammar and vocabulary. Therefore, they require services provided with sign language, but there are only a few services provided. For example, less than 0.5% of airtime of TV programs have sign language services in Japan (Ministry of Internal Affairs and Communications, 2019). There are many efforts to develop systems to provide more services in sign language through computer graphics (CG)-based animation (Kipp et al., 2011; Romeo et al., 2014; Uchida et al., 2018; Azuma et al., 2018). These systems are designed for practical domain-specific services. Therefore, they apply rule-based translation methods. Typical rule-based translation methods can translate with high quality for the target domain, but the coverage for the input tends to be narrow.

To provide sign language services that can be used for flexible, open-domain target contents, non-rule-based machine translation is necessary. Machine translations generally require large-scale training corpora (Koehn and Knowles, 2017; Lample et al., 2018). However, there are only small corpora for sign languages; one reason is that sign languages do not have writing systems. To overcome this data-shortage scenario, we use a pre-trained language model of spoken language for the machine translation model. Our method is based on Transformer (Vaswani et al., 2017), and we use BERT (Devlin et al., 2019) as the initial model of the encoder. The encoder embeds the input transcribed spoken language, then the embedded vectors are fed to the decoder, which is also based on Transformer, then the input sentences are translated into sign language glosses. Evaluation results indicate that our method outperformed baseline methods, including phrase-based statistical machine translation (PBSMT)-based method, using only 130,000 sentence pairs of training data. We also show that one of the reasons of translation error is from Pronoun, which is a special feature used in sign language. We also conducted trials to improve the translation quality for Pointing. The results are somewhat disappointing, so we believe that there is still room for improving translation quality, especially for Pointing.

2. Related Work
2.1. Sign Language Translation
Statistical machine translation (SMT) methods are widely used, so many studies on sign language translation are based on such methods. Stein et al. (2010) applied many SMT techniques and obtained high translation quality with a small corpus. San-Segundo et al. (2012) reported on combining three translation methods — example-based, rule-based, and SMT — to translate from spoken Spanish to sign language. There are several methods that adopt special features of sign language such as mouthing, facial expression, and expression speed. Massó and Badia (2010) used these special features for training data and obtain good results. Morrissey (2011) used HamNoSys (Hanke, 2004) as a sign language translation method, which can be expanded by taking into account not only the word meanings but also facial and other expressions.
### 3. Our Corpus

#### 3.1. Corpus Overview

We have been building a Japanese-Japanese Sign Language (JSL) news corpus to study Japanese to JSL machine translation. The corpus was created from daily NHK sign language news programs, which are broadcast on NHK TV with Japanese narration and JSL signing.

| Feature   | Description                                                                 | Freq. |
|-----------|-----------------------------------------------------------------------------|-------|
| Nodding   | Used as punctuations, topicalization, and conjunctions.                      | 4.91  |
| Pointing  | Typically used as pronouns, but also used as emphasizing the meanings and indicating the former word as subject of the sentence. | 1.75  |
| Classifier| Morphological system that can express events and states using many morpheme. | 0.26  |

Table 1: Special features of JSL transcribed in the corpus. Freq. represents frequency in the corpus (number of features per sentence).

| JP         | 東京は夜から雪や雨の降る所がある見込みです。 |
|------------|----------------------------------------|
| EN         | Tokyo will have places where snow and rain will fall from tonight.          |
| SL         | Nodding, TOKYO, R: TOKYO + L:Pointing, DARK, FROM, Nodding, SNOW, Nodding, RAIN, Nodding, REGION, EXIST, DREAM, Nodding |

Table 2: Examples from our corpus. JP means Japanese transcription, EN means translation of JP into English, and SL means sign language word sequences, with word segmentation of “,”.

The corpus consists of Japanese transcriptions, JSL videos, and JSL transcriptions. Japanese transcriptions were transcribed by revising the speech recognition results of news programs. JSL transcriptions are carried out by changing the sign motions of the newscasters into sign word glosses. The JSL videos were manually extracted from the programs by referring to the time intervals of the transcribed JSL transcriptions. The corpus currently includes about 130,000 sentence pairs taken from broadcasts running from April 2009. In this corpus, sign languages are presented by 18 casters (11 deaf casters and 7 hearing-able interpreters). Note that, Japanese and JSL phrase pairs are not literal translations, so there are many subject complements, omissions, and so on.

#### 3.2. Sign Words Transcription Rules

JSL transcriptions of the corpus were manually transcribed by native JSL speakers. The words in the transcriptions are represented using the Japanese words that have the most similar meanings. We also transcribed the special features listed in Table 1, which are frequently used in JSL.

This notation method is called “glosses” in sign language research. Examples from our corpus are shown in Table 2. Note that, our transcription also includes multi-linear expressions, such as place name using the right hand and pointing with the left hand at the same time. For example in Table 1, “R:TOKYO + L:Pointing” means the place...
name “Tokyo” is expressed with the right hand, and Pointing with left hand at the same time. We use only sign word sequences expressed using the right hand in this paper.

4. Translation with Pre-trained Model

As we mentioned in Section 3.1., we have only 130,000 sentence pairs in our corpus. This is far smaller than open corpora used in machine translation such as the WMT 2014 English–German dataset, which contains around 4.5M sentence pairs. Generally, sign languages do not have writing systems, so transcriptions of sign language are difficult to gather.

To overcome the shortage of training data, we use a pre-trained model as the initial model of the encoder of the translation model. An overview of our method is illustrated in Figure 1. Our method is based on Transformer (Vaswani et al., 2017) and uses a pre-trained BERT model (Devlin et al., 2019) as the initial model of the encoder. Input sentences written in spoken language are embedded using the encoder, then the embedded vectors are fed into the decoder and translated into sign language glosses. The learning process involves fine-tuning the pre-trained model and learning the decoder in parallel.

The pre-trained model can embed input Japanese sentences more relevantly than that learned from a parallel corpus, so it can help improve overall translation quality. Moreover, most of the “loss” calculated in the training process can be used to optimize the decoder due to the difference in the training rate between the encoder and decoder, so training the decoder can progress rapidly. We call our method “NMT-BERT.” Our translation model is almost the same as that Imamura and Sumita (2019) used. Our study differs in that we applied the model to sign language.

There are many techniques to improve translation models such as tied embedding, label smoothing, and data augmentation. However, we did not use them because we wanted to confirm the effectiveness of the pre-trained model in translation.

5. Experiment

5.1. Experimental Settings

Our experiments were based on our corpus mentioned in Section 3. We randomly divided the corpus into 130,215 sentence pairs for training, 1,000 pairs for development, and 2,000 pairs for testing. We also prepared reduced training datasets containing 50,000, 10,000, and 1,000 sentence pairs for comparing performance in low-data settings. We denote the 130,215 sentence pairs of training data as 130K, that of 50,000 as 50K, 10,000 as 10K, and 1,000 as 1K.

For the encoder of our method, we used our in-house Japanese BERT model learned from about 7.1 GB of Japanese Wikipedia, Twitter, News articles, and other corpora. Hyperparameters were the same as BERT-base\(^1\), which has a 12-layer, 768 hidden states Transformer model with 12-head attention. We used SentencePiece (Kudo and Richardson, 2018) as the tokenizer for Japanese with a vocabulary size of 32,000.

For the decoder, we used Transformer, which has 768 hidden states, with 8-head attention with layer normalization for each layer, and the number of layers are four for all training data, which are tuned based on the BLEU score (Papineni et al., 2002) on the development data. We used beam search in translating with a beam size of 10. We used each JSL word as a token, and words used less than 5 times in the corpus were regarded as out-of-vocabulary words (OOVs). As a result, the decoder has a vocabulary size of 5,984.

The models were implemented using pytorch\(^2\) with Transformers\(^3\) and learned with the Adam optimizer (Kingma and Ba, 2015) on the basis of cross-entropy loss. We used the stochastic gradient descent with warm restarts (SGDR) scheduler (Loshchilov and Hutter, 2017) without restart to adjust the learning rate with 5 epochs for warmup. The learning rates were \(1.0 \times 10^{-3}\) for training the decoder and \(2.0 \times 10^{-5}\) for fine-tuning the pre-trained model.

Other hyperparameters used were: a minibatch size of 50, dropout rate of 0.1, and 50 training iterations with early stopping on the basis of the BLEU score for the development data.

5.2. Baseline Methods

5.2.1. PBSMT Baseline

We prepared the phrase-based statistic machine translation (PBSMT) baseline method. We used Moses v4 (Koehn et al., 2007) to train for this baseline. We used MGIZA (Gao and Vogel, 2008) for word alignment and Implz of KenLM (Heafield et al., 2013) for 5-gram language model training. We also used batch MIRA (Cherry and Foster, 2012) to optimize feature weights on the development data with the target metric of the BLEU score. We denote this method as “PBSMT.”

\(^1\)https://github.com/google-research/bert
\(^2\)https://pytorch.org/
\(^3\)https://github.com/huggingface/transformers
5.2.2. Transformer without Pre-trained Model
We used Transformer without a pre-trained model as another baseline. With this method, all parameters are trained from training data. The number of layers for the encoder and decoder were two for all training data, and other hyper-parameters were the same as NMT-BERT, which were the best settings on the development data. We also used SentencePiece as a tokenizer with the model learned from the training corpus with a vocabulary size of 8,000. We denote this method as “NMT-Base.”

5.3. Results
Table 3 presents the experimental results. NMT-BERT outperformed the baseline methods for 130K and outperformed NMT-Base for all datasets. Therefore, we confirmed that using a pre-trained model for NMT is effective especially in small-training-data situations. However, PBSMT is still the best for smaller datasets. NMT-BERT outperformed NMT-Base, especially for low-training-data situations. Generally, learning an NMT models requires large-scale parallel corpora. However, NMT-BERT requires only a small parallel corpus and large-scale monolingual corpus of spoken language, which are rather easy to create. This is very advantageous, especially for sign language translation, because corpora of sign language are difficult to create.

Figure 2 shows the BLEU scores for the development data for each epoch. We show two cases for the training datasets, 130K and 10K. NMT-BERT was far better in early learning processes (around epochs 1–10). The encoder learned only from the training data outputting almost random vectors in the early epochs, but the pre-trained model could output relevant vectors for the input sentence. The decoder of NMT-BERT can use these relevant vectors, so optimizing the decoder can be easier than that of NMT-Base. Moreover, the output vector of the pre-trained model represents word-to-word relations such as synonym and paraphrases, so NMT-BERT can translate OOVs or less frequent words in the training corpus using these relations. This is why NMT-BERT outperformed NMT-Base.

On the other hand, PBSMT was best for 10K and 1K. The pre-trained model is useful for improving translation quality, but there is a limit. For these very small training data situations, other techniques such as that used by Sennrich and Zhang (2019) should be used. Sign languages have special features such as Nodding and Pointing. We analyzed our translation results to investigate the effect of these special features. Table 4 shows the BLEU score of excluding Nodding or Pointing from both translation results and reference data for NMT-BERT.

The fact that excluding Pointing increases the BLEU score by around 1.0 suggests that translating Pointing is difficult. Pointing is typically used as pronouns but sometimes used to emphasize the meanings of nouns or indicate the word as the subject of the sentence, so spoken languages do not have the same word/function. This is why Pointing is difficult to translate. On the other hand, excluding Nodding lowers the BLEU score. Nodding is mostly used as punctuations, topicalization, and conjunctions. These functions are also used in spoken language, so Nodding can be translated easily.

5.4. Toward Improving Pointing Translation
To improve Pointing translation, we evaluated three translation methods. One involves translating Pointing as a sign word, i.e., the same as with NMT-BERT, and is denoted as Translating (Figure 3-(a)). The second method combines Pointing and the former word into one token and is denoted as Jointed-Pointing (Figure 3-(b)). If the meanings of Pointing are decided only by the former word, combining Pointing and the former word can clarify their meanings, so it may help improve translation quality. The third method involves using sequential labeling for Pointing and is denoted as Sequential labeling (Figure 3-(c)). Sequential labeling is typically used for finding specific parts from a sequence such as named entity recognition or part-of-speech tagging by taking into account context and grammatical rules (Ma and Hovy, 2016). We used this method to find the specific part—to use Pointing—from the sentence. If the decision to use Pointing is made by context and grammatical rules rather than the former word, Sequential labeling will work well. With Sequential labeling, we use multi-task learning for two tasks—translating into sign language and judging whether pointing is needed for the translated word. —

We did not analyze for Classifier, which is a special features of sign language. This is because Classifier plays an important role for the meanings of a sentence, so excluding classifier make a sentence meaningless, so the evaluation would not make sense.
We showed that Transformer with a pre-trained model can be used with a small amount of training data, so we can apply many techniques designed for use with Transformer such as tied embedding, label smoothing, and data augmentation. Using these techniques is for our future work. To improve the translation quality of special features of sign language such as Pointing is also for future work.

**Acknowledgements**

We would like to thank the anonymous reviewers for their helpful comments.

7. Bibliographical References

Azuma, M., Hiruma, N., Sumiyoshi, H., Uchida, T., Miyazaki, T., Umeda, S., Kato, N., and Yamanouchi, Y. (2018). Development and evaluation of system for automatically generating sign-language CG animation using meteorological information. In *International Conference on Computers Helping People with Special Needs*, pages 233–238. Springer.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Cihan Camgoz, N., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Cormier, K., Schembri, A., and Woll, B. (2013). Pronouns and pointing in sign languages. *Lingua*, 137:230–247.

Dabre, R., Fujita, A., and Chu, C. (2019). Exploiting multilingualism through multistage fine-tuning for low-resource 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 1410–1416, Hong Kong, China, November. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57.

Hanke, T. (2004). HamNoSys-representing sign language data in language resources and language processing contexts. In *LREC*, volume 4, pages 1–6.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model
estimation. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 690–696.

Imamura, K. and Sumita, E. (2019). Recycling a pre-trained BERT encoder for neural machine translation. In Proceedings of the 3rd Workshop on Neural Generation and Translation, pages 23–31.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations 2015.

Kipp, M., Heloir, A., and Nguyen, Q. (2011). Sign language avatars: Animation and comprehensibility. In International Workshop on Intelligent Virtual Agents, pages 113–126. Springer.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. ACL 2017, page 28.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pages 177–180.

Kudo, T. and Richardson, J. (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, pages 66–71.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5039–5049.

Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In International Conference on Learning Representations 2017.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1064–1074.

Massó, G. and Badia, T. (2010). Dealing with sign language morphemes in statistical machine translation. In 4th workshop on the representation and processing of sign languages: corpora and sign language technologies, Valletta, Malta, pages 154–157.

Ministry of Internal Affairs and Communications. (2019). Achievements of closed caption etc. for TV programs in 2018. Press release (in Japanese).

Mocialov, B., Hastie, H., and Turner, G. (2018). Transfer learning for British Sign Language modelling. In Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018), pages 101–110, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Morrissey, S. (2011). Assessing three representation methods for sign language machine translation and evaluation. In Proceedings of the 15th annual meeting of the European Association for Machine Translation (EAMT 2011), Leuven, Belgium, pages 137–144.