Approaching Sign Language Gloss Translation as a Low-Resource Machine Translation Task

Xuan Zhang xuanzhang@jhu.edu
Kevin Duh kevinduh@cs.jhu.edu
Johns Hopkins University, Baltimore, MD 21218, USA

Abstract

A cascaded Sign Language Translation system first maps sign videos to gloss annotations and then translates glosses into a spoken languages. This work focuses on the second-stage gloss translation component, which is challenging due to the scarcity of publicly available parallel data. We approach gloss translation as a low-resource machine translation task and investigate two popular methods for improving translation quality: hyperparameter search and back-translation. We discuss the potentials and pitfalls of these methods based on experiments on the RWTH-PHOENIX-Weather 2014T dataset.

1 Introduction

More than 70 million deaf people around the world use sign language as their primary language to communicate. In a society dominated by hearing people and spoken languages, there is a risk that deaf people may experience inconvenience and isolation. In countries like India, Iran, and Russia, lack of sign language interpreters hampers access to public services and courts (Kozik, 2019). Automatic Sign Language Translation (SLT) has recently gained increasing attention from researchers and would help remove the communication barriers.

Sign language is not simply a visual form of spoken languages. It has its own linguistic rules, including phonology, morphology, syntax and semantics that are different from other languages (Valli et al., 2011). For example, in American Sign Language, the subject or object might be omitted in certain situations. There is also a process called *Topicalization*, where prominent information is signed first, resulting in an adjustment to the basic SVO word order. Linguists use *glossing* to annotate signs, which can be viewed as a written form of sign language. Glosses can be taken as intermediate representations when translating continuous sign utterances to spoken language sentences.

Previous work on SLT adopts either an end-to-end system that maps sign language videos directly to spoken languages, or a cascaded system, as shown in Figure 1, that first relies on Continuous Sign Language Recognition (CSLR) to produce sign glosses and then passes the glosses into a Neural Machine Translation (NMT) system (Camgoz et al., 2018, 2020; Yin and Read, 2020). Importantly, Camgoz et al. (2018) reports that the cascaded system outperforms the end-to-end system by a large margin (18.13 vs. 9.58 BLEU). In this work, we focus on improving the NMT component of cascaded systems, which attracts much less attentions compared to the CSLR component of cascaded systems (Cui et al., 2017; Huang et al., 2018; Yang et al., 2019; Orbay and Akarun, 2020).

Sign language gloss translation is a challenging problem due to the scarcity of annotated parallel data. The popular continuous SLT dataset, "RWTH-PHOENIX-Weather 2014T" (Camgoz et al., 2018) contains 7,096 gloss-text examples in training set. However, the state-of-the-art

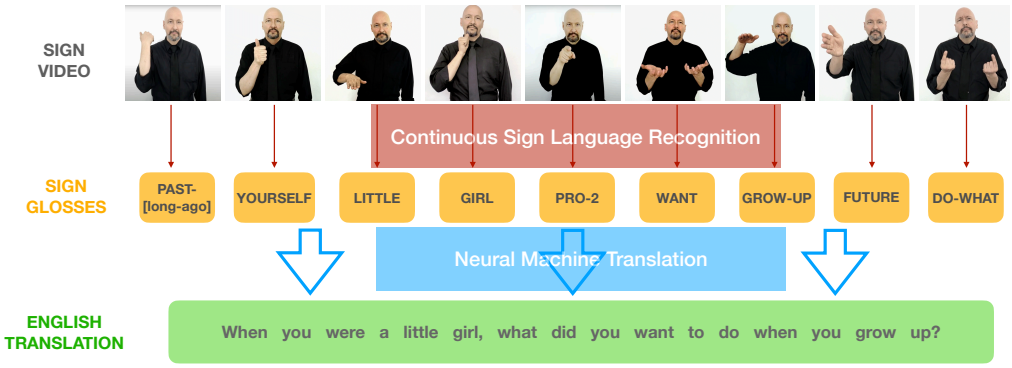

Figure 1: Cascaded sign language translation system[2]- First, CSLR converts sign video to a sequence of sign glosses. Then, NMT converts the sign glosses to text in e.g. English.

NMT systems are known to be extremely data-hungry and usually require millions of training examples to obtain a good translation performance (Koehn and Knowles, 2017). In this paper, we approach gloss to text translation as a low-resource machine translation task and investigate two methods that are widely explored by the machine translation community to alleviate the need of large corpora, namely hyperparameter search and back-translation.

While NMT models, like Transformer (Vaswani et al., 2017) can perform well with default hyperparameter settings on most of the publicly available large corpora, its performance is highly sensitive to hyperparameters under low-resource scenarios (Araabi and Monz, 2020; Duh et al., 2020). The optimal hyperparameter settings for a large corpus might lead to a poor system trained on a small dataset (Zhang and Duh, 2020). In this work, we focus on tuning 4 hyperparameters and find that hyperparameter search is necessary and helpful for gloss translation.

Back-translation (Sennrich et al., 2016a) incorporates monolingual data in NMT which can help in low-resource settings (Hoang et al., 2018; Lample et al., 2018; Feldman and Coto-Solano, 2020). Our experiments show that it has potential on gloss translation when the additional monolingual data are from the same domain as the parallel data.

Overall, we conclude that the low-resource machine translation perspective is promising but should not be taken as the ultimate solution for sign language gloss translation. It may be more promising to first focus on creating larger gloss-text datasets.

2 Related Work

Most of the Sign Language Processing research has focused on Sign Language Recognition (Yin et al., 2016; Wang et al., 2016; Camgöz et al., 2016; Vaezi Joze and Koller, 2019). Recent work started to show an interest in CLSR (Koller et al., 2016; Cui et al., 2017; Huang et al., 2018; Yang et al., 2019; Orbay and Akarun, 2020). However, only a few works move forward to tackle this problem as a SLT task. Camgoz et al. (2018) formalized SLT in the framework of NMT and released the first publicly available SLT dataset, PHOENIX14T. Based on this dataset, Camgoz et al. (2020) and Yin and Read (2020) explored SLT with Transformers and developed both end-to-end and cascaded systems where gloss annotations are used as intermediate representations. Ko et al. (2019) proposed a sign language translation system based on human keypoint estimation and also introduced the KETI dataset, which consists of Korean

[2]Sign videos and glosses are from `lifeprint.com`.

sign videos and annotations. KETI has only 105 gloss annotations. Also, the sentences in KETI are relatively short as they are related to emergency situations. Othman and Jemni (2012) introduced the ASLG-PC12 dataset, which consists of millions of English sentences and corresponding American sign language glosses. However, the glosses are generated by applying transformation rules on English sentences and are not reliable for the study of SLT.

3 Data and Setup

We evaluate the effectiveness of hyperparameter search on a parallel gloss-text dataset. For back-translation, we experiment with monolingual data from the same domain as the parallel data and data from a different domain respectively. In this section, we will describe the datasets and NMT models in detail. We will also introduce our experimental setup.

3.1 Parallel Data

We use "RWTH-PHOENIX-Weather 2014T" introduced by Camgoz et al. (2018) as our parallel gloss-text dataset. PHOENIX14T collected the weather forecast airings of the German public tv-station PHOENIX. It is a continuous SLT corpus, which contains sign videos, gloss annotations and German translations. The data split for train/dev/test is 7,096/519/642 sentences. The vocabulary size of the training set for glosses and German[3] are 1,066 and 2,887 respectively.

3.2 Monolingual Data

To the best of our knowledge, there is no publicly available large corpus of weather forecast subtitles in German. Since domain mismatch between the monolingual data and the parallel data might hurt the performance of NMT systems (Koehn and Knowles, 2017), we adopt several domain adaption methods to alleviate this problem. We use Moore-Lewis filtering (Moore and Lewis, 2010) to select sentences similar to PHOENIX14T from a German TED Talk corpus (Duh, 2018), which consists of 151,627 sentences.

3.3 NMT Model

Most NMT models in literature follow a encoder-decoder architecture. The conditional probability of generating the target sentence $\boldsymbol{y}$ given the source sentence $\boldsymbol{x}$ is decomposed as:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{J} p(y_j \mid \boldsymbol{y}_{<j}, \boldsymbol{x}, \theta), \tag{1}$$

where θ represents model parameters, y_j is the j-th target word, and $\boldsymbol{y}_{<j}$ is the prefix of words before y_j. The encoder of an NMT model transforms $\boldsymbol{x}$ into a sequence of hidden states, the decoder then generates y_j iteratively based on the hidden states and the history decoding states to form the target sentence $\boldsymbol{y}$. We choose Transformer (Vaswani et al., 2017) as it is the de facto mainstream NMT architecture and has achieved the state-of-the-art performance on many machine translation tasks. Transformer is an encoder-decoder based model with each layer consisting of a multi-head attention mechanism, followed by a feed-forward network.

3.4 Experimental Setup

3.4.1 Data Preprocessing

All datasets are tokenized using the Moses (Koehn et al., 2007) tokenizer. We train the Byte-Pair-Encoding (BPE) segmentation (Sennrich et al., 2016b) models separately for gloss and text. For hyperparameter search experiments (Section 4), we learn BPE models from PHOENIX14T. For back-translation tasks (Section 5), on the German side, we learn BPE models from the

[3]For the rest of this paper, we will refer to sign language gloss as gloss and the spoken German as German.

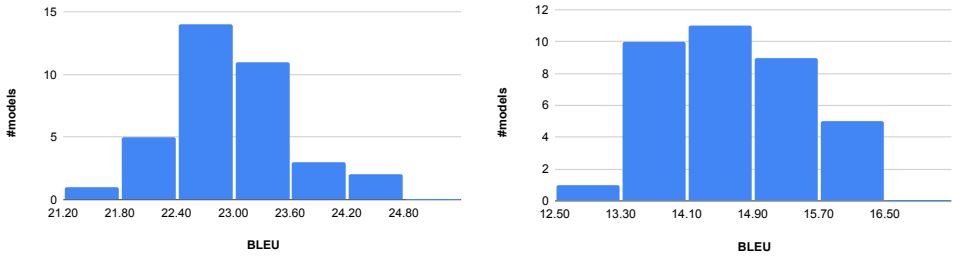

Figure 2: Histograms of BLEU scores showing wide variance in performance with different hyperparameter settings trained on PHOENIX14T gloss-text (left) and text-gloss (right).

concatenation of selected monolingual data and PHOENIX14T data. On the gloss side, we learn BPE models from the concatenation of PHOENIX14T data and the back-translated TED Talk glosses.

3.4.2 NMT Setup

Our NMT models are developed in Sockeye[4] (Hieber et al., 2017). The number of attention heads is set to 8 and the feed-forward layer dimension is set to 1024. We set the dropout probability for source and target embeddings to 0.1. Our models apply Adam (Kingma and Ba, 2014) as the optimizer. The learning rate is multiplied by 0.9 whenever validation perplexity does not surpass the previous best in 8 checkpoints, where checkpoints are encountered for every 1000 updates/batches. And each batch consists of 2048 words. Training stops when the perplexity on the development set has not improved for 32 checkpoints.

All the back-translation experiments in Section 5 adopt the best hyperparameter settings obtained by a hyperparameter search (Section 4). Note that the optimal settings for gloss-text translation is different from text-gloss translation.

4 Hyperparameter Search

Hyperparameter selection is crucial to build a good NMT system. It is especially the case for low-resource scenarios when the default hyperparameter settings are very likely to be ineffective. As reported in Sennrich and Zhang (2019) and Zhang and Duh (2020), the NMT systems developed for low-resource translation tasks disagree a lot with those trained on high-resource corpora on the optimal hyperparameter choices. Furthermore, datasets in different domains and language pairs all differ in their hyperparameter preference. They also show that adjusting hyperparameters can cause BLEU differences of more than 20 in some datasets.

4.1 Important Hyperparameters

In this work, we focus on 4 hyperparameters of Transformer models: the number of BPE merge operations, the number of layers, embedding dimensions and initial learning rate. These hyperparameters are recognized as important hyperparameters by Zhang and Duh (2020), where the importance is computed as the variation in BLEU when changing a specific hyperparameter with values of all the other hyperparameters fixed (Klein and Hutter, 2019).

BPE is a word segmentation approach that combines frequent sequence of characters so that out-of-vocabulary words are handled. It is expected to improve the translation of rare words and has been a standard preprocessing practice in NMT. According to Ding et al. (2019), although 32k and 90k are popular choices in most machine translation literature, they found

[4] github.com/awslabs/sockeye

	gloss-text					text-gloss				
	bpe	#layer	#embed	init_lr	**BLEU**	bpe	#layer	#embed	init_lr	**BLEU**
best	1k	4	512	0.00005	**24.38**[5]	1k	4	256	0.0005	**16.43**
worst	2k	1	512	0.0005	**21.73**	1k	1	512	0.0005	**13.04**
random	1k	2	256	0.0002	**23.49**	1k	2	256	0.0002	**15.74**

Table 1: Performance of selected Transformers. BLEU scores are evaluated on the test set of PHOENIX14T. **Best** and **worst** are best and worst systems obtained from hyperparameter search respectively. **Random** randomly picks a hyperparameter setting from our search space.

that the BPE of the best Transformer-based architectures in low-resource setting is somewhere between 0-2k. We thus try 1k and 2k in our experiments.

Architecture design hyperparameters like the number of layers in encoder and decoder and embedding size are important. A big and complex model is more susceptible to overfitting. On the other hand, if the model is too small and simple, it might struggle to capture the meaningful patterns of data and result in underfitting. Our search space includes 1, 2, 4 layers and embedding size of 256 and 512.

The learning rate is another important hyperparameter that scales the gradient in gradient descent training. A small initial learning rate may prolong the training process, whereas a large one may get the model stuck in a sub-optimal solution. It is recommended to start training with a low number (Koehn, 2020). We adjust it among 0.00005, 0.0002 and 0.0005.

We tune hyperparameters for NMT systems on both gloss-text and text-gloss directions. This sums up to 72 systems in total.

4.2 Results

The BLEU scores obtained on our search space are illustrated in Figure 2, where a wide variance is observed. As shown in Table 1, different choices of hyperparameters can increase the BLEU score by as much as 2.65 on gloss-text and 3.39 on text-gloss. Training is not expensive due to the small data size, so running a wide search over hyperparameters is recommended.

5 Back-translation

Back-translation proposed in Sennrich et al. (2016a) has shown its effectiveness in utilizing monolingual data to improve the translation performance. It is particularly used in low-resource scenarios. When it comes to sign language translation, the written text is always abundant, whereas the glosses and parallel examples are expensive to get and are not sufficient to train a robust NMT model. This sets a good stage for back-translation.

The workflow of back-translation is illustrated in Figure 3. In order to train a more robust gloss-text translation model, one first trains a text-gloss model using the PHOENIX14T parallel data (Figure 3, step 1). This model is then employed to translate monolingual German text in the domain of TED Talk to glosses (Figure 3, step 2). This synthetic parallel corpus is then concatenated with the PHOENIX14T data to train the final gloss-text system (Figure 3, step 3).

One problem with our implementation of back-translation is that TED Talk subtitles have different styles compared to PHOENIX14T, which is composed of weather reports, in other words, they are in different domains. Domain mismatch makes the translation task even more challenging, as the synthetic parallel data might introduce noises to hurt the performance. In order to alleviate this issue, we adopt two domain adaptation methods to aid back-translation.

[5]The best BLEU-4 score reported in Camgoz et al. (2020) and Yin and Read (2020) are 24.54 and 24.9, which are comparable to our results.

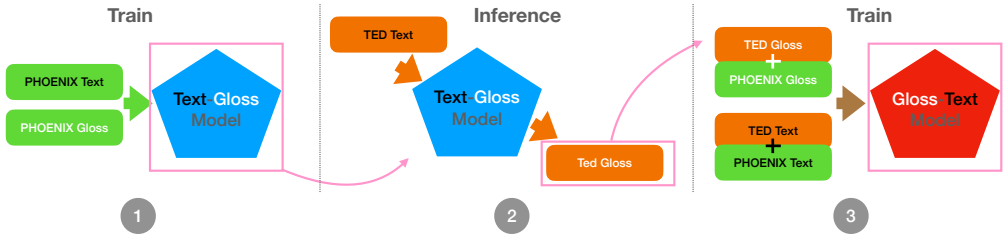

Figure 3: Back-translation workflow. Pink rectangles frame outputs of each step.

5.1 Domain Adaptation Methods

Domain adaptation leverages out-of-domain data to improve the domain-specific translation. In this work, TED Talks are out-of-domain data and PHOENIX14T is the domain in interest or in-domain data. We adopt a language model based data selection method to first select examples from TED Talk corpus that are similar to PHOENIX14T. Next, those monolingual examples are back-translated and form a synthetic parallel corpus. A model is then trained on the concatenation of the synthetic data and the parallel PHOENIX14T data. This is not the end. Finally, we continue training the model with only the PHOENIX14T data. This continued-training process is also called fine-tuning, which is the conventional way for domain adaptation (Luong et al., 2015; Sennrich et al., 2016a; Chu and Wang, 2018; Zhang et al., 2019).

5.1.1 Data Selection

We adopt the data selection method proposed in Moore and Lewis (2010). The main idea is to score the out-of-domain data N using language models trained from the in-domain data I and N and select top n training examples from N by a cut-off threshold on the resulting scores. To be specific, each sentence s in N is assigned a cross-entropy difference score,

$$H_I(s) - H_N(s), \tag{2}$$

where $H_I(s)$ is the per-word cross-entropy of s according to a language model trained on 1000 random samples of PHOENIX14T, and $H_N(s)$ is the per-word cross-entropy of s according to a language model trained on 1000 random samples of TED Talks. A lower score indicates s is more like a sentence in weather forecast then in TED Talks.

5.1.2 Fine-tuning

In conventional fine-tuning, a NMT model is trained on a high-resource out-of-domain corpus until convergence, and then its parameters are fine-tuned on a low-resource in-domain corpus. We approach it in a slightly different way. Instead of training on out-of-domain corpus at the first step, we train on a shuffled combination of both in-domain and out-of-domain data, where the small-sized in-domain data may be copied several times, and the size of the out-of-domain data subset varies across experiments. This data size variation is intended to help us explore how different weighting and combination of data impacts final results.

5.2 Experimental Comparison

In order to evaluate the effectiveness of back-translation on low-resource gloss-text translation, we conduct experiments enhanced with data selection and fine-tuning techniques. We investigate the effect of data ratio by varying both the size of monolingual TED Talk data and the size of PHOENIX14T. For TED Talks, we adjust the cut-off threshold of the data selection score and result in 10k, 50k and 100k most relevant examples. For PHOENIX14T, as it is a tiny dataset,

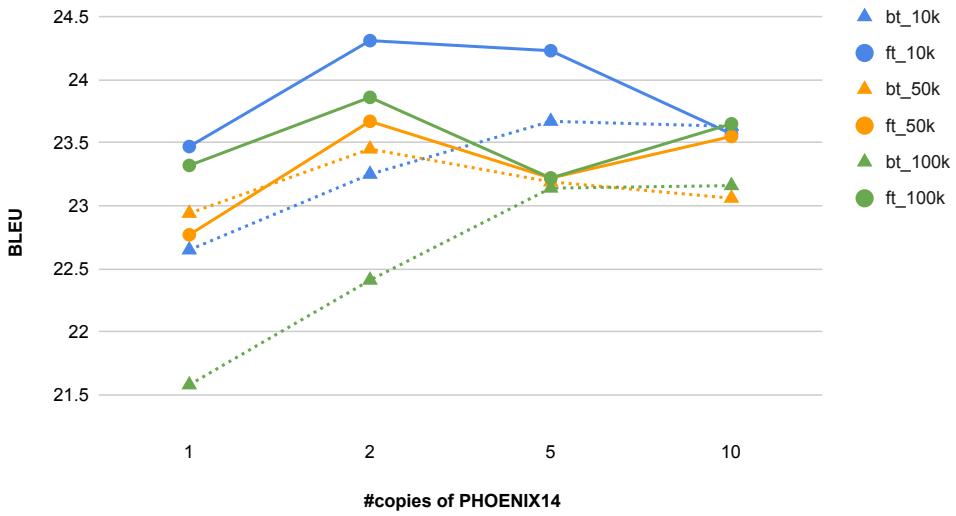

Figure 4: Performance of NMT systems on gloss-text translation. BLEU scores are evaluated on the test set of PHOENIX14T. Systems vary in whether fine-tuned on PHOENIX14T (bt vs. ft), the size of synthetic TED Talk data (10k, 50k, 100k) and the number of copies of PHOENIX14T added into the training data (0, 1, 2, 5, 10).

we simply balance the data ratio by making 0, 1, 2, 5 or 10 copies of it to combine with the synthetic TED Talk data. This new corpus is used to train a gloss-text system, which we call **bt** systems. Those systems are then fine-tuned on PHOENIX14T and result in **ft** systems.

5.3 Results on Out-Of-Domain Data Incorporation

We report the performance of different NMT systems in Figure 4.

Effect of fine-tuning Comparing **ft** to **bt** models, **ft** outperforms **bf** 10 out of 12 times. The improvement ranges from **0.03** BLEU to **1.74** BLEU, with an average of **0.7** BLEU. The largest improvement is achieved by **ft_100k_1**[6]. This shows that although the large amount of noisy out-of-domain data hurts the performance of the **bt** system, fine-tuning on only a small amount of in-domain data still improves the performance to a great extent.

Effect of data selection In order to evaluate the influence of data selection, we train two extra gloss-text systems and fine-tune them on PHOENIX14T. The difference is that system 1 uses randomly sampled TED data, while system 2 uses TED data selected by the Moore-Lewis method. It turns out that system 2 outperforms system 1 by 7.28 BLEU. Therefore, data selection is crucial when incorporating out-of-domain data in NMT.

Effect of the amount of out-of-domain data With the size of PHOENIX14T data fixed, **ft_10k** models are overall better than **ft_100k** models, which are better than **ft_50k** models. This is not the case for **bt** models, where **bt_100k** models tend to be worse than **bt_10k** and **bt_50k** models. This reveals one weakness of back-translation – it is prone to the quality of the

[6]This is short for a fine-tuned system that was trained on a concatenation of 100k TED data and 1 copy of PHOENIX14T data.

NMT System	BLEU
gloss-text_part1	19.13
text-gloss_part1	9.96
gloss-text_part1+synthetic part2	21.57

Table 2: Performance of NMT systems with in-domain monolingual data incorporation.

synthetic parallel data..

Effect of the amount of in-domain data Increasing the ratio of in-domain data in training data is not always beneficial – too much in-domain data might even hurts the performance. In practice, the data ratio can be taken as a tunable parameter and choose it wisely.

Effect of back-translation The best system enhanced with back-translation and domain adaptation techniques achieves **24.31** BLEU, which is slightly worse than the best BLEU score (**24.38**) achieved by hyperparameter search (Table 1). We wonder whether it would make a difference if the domain issue is eliminated. In next section, we simulate a condition when extra in-domain monolingual examples are available.

5.4 Results on In-Domain Data Incorporation

In order to evaluate back-translation on a less simpler situation, where domain mismatch is not a concern, we divide the PHOENIX14T training set into 2 parts. Each part contains 3,548 samples. We treat part 1 as a parallel corpus, while for part 2, we discard all the glosses and only keep the German text to simulate additional in-domain monolingual data.

We first train a gloss-text system on part 1 as a baseline (**gloss-text_part1**). Next, we train a text-gloss system on part 1 (**text-gloss_part1**). We then use this system to translate the German text from part 2 into synthetic glosses. The final gloss-text system is trained on the concatenation of part 1 and the synthetic parallel data of part 2 (**text-gloss_part1+synthetic part2**). The performance of the 3 systems on PHOENIX14T test set is shown in Table 2. The resulting gloss-text system improves over the baseline system by a margin of 2.44 BLEU. We can expect that given high-quality in-domain monolingual data, back-translation still has a great potential in improving the translation quality.

6 Conclusions

In this paper, we identify one challenging task in Sign Language Translation, that is the translation between sign language glosses and written languages. We argue that the obstacle lies in the sparsity of parallel data. To conquer this problem, we propose to approach sign language gloss translation as a low-resource machine translation task. We investigate the effectiveness of hyperparameter search and back-translation, which are both widely used by machine translation community for low-resource translations. We conclude that hyperparameter search is necessary, whereas back-translation is susceptible to the quality of additional monolingual data. If there is abundant in-domain monolingual data, back-translation is very promising. Otherwise, it should be used with domain adaptation techniques, like data selection and fine-tuning to achieve a reasonable performance.

Given limited parallel data, the upper bound of these low-resource methods is constrained. We thus urge the sign language processing community to put in extra efforts in creating more annotated parallel data.

References

Araabi, A. and Monz, C. (2020). Optimizing transformer for low-resource neural machine translation. *arXiv preprint arXiv:2011.02266*.

Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Camgöz, N. C., Kındıroğlu, A. A., and Akarun, L. (2016). Sign language recognition for assisting the deaf in hospitals. In Chetouani, M., Cohn, J., and Salah, A. A., editors, *Human Behavior Understanding*.

Camgoz, N. C., Koller, O., Hadfield, S., and Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chu, C. and Wang, R. (2018). A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Cui, R., Liu, H., and Zhang, C. (2017). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ding, S., Renduchintala, A., and Duh, K. (2019). A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*.

Duh, K. (2018). The multitarget ted talks task. `http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/`.

Duh, K., McNamee, P., Post, M., and Thompston, B. (2020). Benchmarking neural and statistical machine translationon low-resource african languages. In *Proceedings of the Language Resources and Evaluation Conference*.

Feldman, I. and Coto-Solano, R. (2020). Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A., and Post, M. (2017). Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Hoang, V. C. D., Koehn, P., Haffari, G., and Cohn, T. (2018). Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Huang, J., Zhou, W., Zhang, Q., Li, H., and Li, W. (2018). Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Klein, A. and Hutter, F. (2019). Tabular benchmarks for joint architecture and hyperparameter optimization. *arXiv preprint arXiv:1905.04970*.

Ko, S.-K., Kim, C. J., Jung, H., and Cho, C. (2019). Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.

Koehn, P. (2020). *Neural Machine Translation*. Cambridge University Press.

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 ACL on interactive poster and demonstration sessions*.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Koller, O., Zargaran, S., Ney, H., and Bowden, R. (2016). Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *BMVC*.

Kozik, K. (2019). Without sign language, deaf people are not equal.

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, Brussels, Belgium. Association for Computational Linguistics.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*.

Orbay, A. and Akarun, L. (2020). Neural sign language translation by learning tokenization. *arXiv preprint arXiv:2002.00479*.

Othman, A. and Jemni, M. (2012). English-asl gloss parallel corpus 2012: Aslg-pc12. In *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon LREC*.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Vaezi Joze, H. and Koller, O. (2019). Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*.

Valli, C., Lucas, C., Mulrooney, K. J., and Rankin, M. N. (2011). *Linguistics of American Sign Language: an introduction*. Gallaudet University Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

Wang, H., Chai, X., Hong, X., Zhao, G., and Chen, X. (2016). Isolated sign language recognition with grassmann covariance matrices. *ACM Trans. Access. Comput.*

Yang, Z., Shi, Z., Shen, X., and Tai, Y. (2019). Sf-net: Structured feature network for continuous sign language recognition. *CoRR*.

Yin, F., Chai, X., and Chen, X. (2016). Iterative reference driven metric learning for signer independent isolated sign language recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *European Conference on Computer Vision 2016*.

Yin, K. and Read, J. (2020). Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Zhang, X. and Duh, K. (2020). Reproducible and efficient benchmarks for hyperparameter optimization of neural machine translation systems. In *Transactions of the Association for Computational Linguistics*.

Zhang, X., Shapiro, P., Kumar, G., McNamee, P., Carpuat, M., and Duh, K. (2019). Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.