Machine Translation between Spoken Languages and Signed Languages Represented in SignWriting

Zifan Jiang
University of Zurich
jiang@cl.uzh.ch

Amit Moryossef
Bar-Ilan University
University of Zurich
amitmoryossef@gmail.com

Mathias Müller
University of Zurich
mmueller@cl.uzh.ch

Sarah Ebling
University of Zurich
ebling@cl.uzh.ch

Abstract

This paper presents work on novel machine translation (MT) systems between spoken and signed languages, where signed languages are represented in SignWriting, a sign language writing system. Our work[1] seeks to address the lack of out-of-the-box support for signed languages in current MT systems and is based on the SignBank dataset, which contains pairs of spoken language text and SignWriting content. We introduce novel methods to parse, factorize, decode, and evaluate SignWriting, leveraging ideas from neural factored MT. In a bilingual setup—translating from American Sign Language to (American) English—our method achieves over 30 BLEU, while in two multilingual setups— translating in both directions between spoken languages and signed languages—we achieve over 20 BLEU. We find that common MT techniques used to improve spoken language translation similarly affect the performance of sign language translation. These findings validate our use of an intermediate text representation for signed languages to include them in NLP research.

1 Introduction

Most current machine translation (MT) systems only support spoken language input and output (text or speech), which excludes around 200 different signed languages used by up to 70 million deaf people[2] worldwide from modern language technology. Since signed languages are also natural languages, Yin et al. (2021) calls for including sign language processing (SLP) in natural language processing (NLP) research.

From a technical point of view, SLP brings novel challenges to NLP due to the visual-gestural modality of sign language and special linguistic features

Figure 1: Demo application based on our models, translating from spoken languages to signed languages represented in SignWriting, then to human poses.

(e.g., the use of space, simultaneity, referencing), which requires both computer vision (CV) and NLP technologies. Crucially, the lack of a standardized or widely used written form for signed languages has hindered their inclusion in NLP research.

However, sign language writing systems do exist and are sporadically used (e.g., SignWriting (Sutton, 1990) and HamNoSys (Prillwitz and Zienert, 1990)). Therefore, we adopt the proposal of Yin et al. (2021) to formulate the sign language translation (SLT) task using a sign language writing system as an intermediate step (illustrated by Figure 1): given spoken language text, we propose to translate to sign language in a written form, then transform this intermediate result into a final video or pose output[3]—and vice versa. According to this multi-step view of SLT, in this work we study translation between signed languages in written form and spoken languages. We use *SignWriting* as the intermediate writing system.

SignWriting has many advantages, like being universal (multilingual), comparatively easy to understand, extensively documented, and computer-supported. In addition, despite looking pictographic, it is a well-defined writing system. Every

---

[1]Code and documentation available at https://github.com/J22Melody/signwriting-translation

[2]According to the World Federation of the Deaf: https://wfdeaf.org/our-work/

[3]Note that the second step, animation of SignWriting into human poses or video, is not included in this research. In the demo application, spoken language text is translated directly into sign language poses, resulting in low-quality output.

sign can be written as a sequence of symbols (box markers, graphemes, and punctuation marks) and their location on a 2-dimensional plane.

To our knowledge, this work is the first to create automatic SLT systems that use SignWriting. Our main contributions are as follows: (a) we propose methods to parse (§3.3), factorize (§3.4), decode (§4.3), and evaluate (§4.3) SignWriting sequences; (b) we report experiments on multilingual machine translation systems between SignWriting and spoken language text (§4); (c) we demonstrate that common techniques for low-resource MT are beneficial for SignWriting translation systems (§5).

2 Background

2.1 Sign language processing (SLP)

SLP (Bragg et al., 2019; Yin et al., 2021; Moryossef and Goldberg, 2021) is an emerging subfield of both NLP and CV, which focuses on automatic processing and analysis of sign language content. Prominent tasks include pose estimation from sign language videos (Cao et al., 2017, 2021; Güler et al., 2018), gloss transcription (Mesch and Wallin, 2012; Johnston and Beuzeville, 2016; Konrad et al., 2018), sign language detection (Borg and Camilleri, 2019; Moryossef et al., 2020), sign language identification (Gebre et al., 2013; Monteiro et al., 2016), and sign language segmentation (Bull et al., 2020; Farag and Brock, 2019; Santemiz et al., 2009).

Besides, tasks including sign language recognition (Adaloglou et al., 2021), translation, and production involve transforming one sign language representation to another or from/to spoken language text, as shown in Figure 2[4]. We find that existing works cover gloss-to-text (Camgöz et al., 2018; Yin and Read, 2020) (where "text" denotes spoken language text), text-to-gloss (Zhao et al., 2000; Othman and Jemni, 2012), video-to-text (Camgöz et al., 2020b,a), pose-to-text (Ko et al., 2019), and text-to-pose (Saunders et al., 2020a,b,c; Zelinka and Kanis, 2020; Xiao et al., 2020).

2.2 Motivation

Our work is the first to explore translation between spoken language text and sign language content represented in SignWriting[5]. We focus on a sign language writing system for the following reasons:
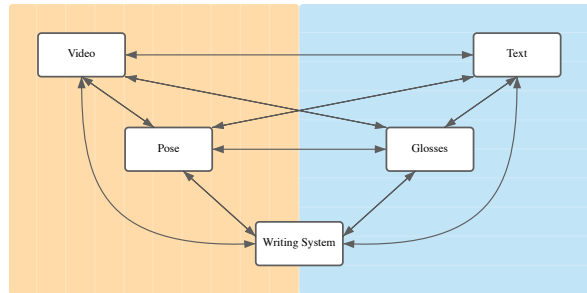

Figure 2: SLP tasks. Every edge on the left side represents a task in CV (language-agnostic). Every edge on the right side represents a task in NLP (language-specific). Every edge crossing both sides represents a task requiring a combination of CV and NLP. Figure taken from Moryossef and Goldberg (2021).

(a) currently an end-to-end (video-to-text/text-to-video) approach is not feasible. State-of-the-art systems either have a BLEU score lower than 1 (Müller et al., 2022a) or work only on a very narrow linguistic domain, e.g., Camgöz et al. (2020b,a) work on the RWTH-PHOENIX-Weather T data set which covers only 1,231 unique signs from weather reports (less than what we use in Table 2); (b) a writing system is lower-dimensional than videos (not all parts of a video are relevant in a linguistic sense), while adequate to encode information of signs; (c) written sign language is a closer fit to current MT pipelines than videos or poses; (d) a phonetic writing system is a more universal solution than glossing since glosses are semantic and therefore language-specific, and are an inadequate representation of meaning (Müller et al., 2022b).

2.3 SignWriting, FSW, and SWU

SignWriting (Sutton, 1990) is a featural and visually iconic sign language writing system (introduced extensively in Appendix A). Previous work explored recognition (Stiehl et al., 2015) and animation (Bouzid and Jemni, 2013) of SignWriting.

SignWriting has two computerized specifications, Formal SignWriting in ASCII (FSW) and SignWriting in Unicode (SWU). SignWriting is two-dimensional, but FSW and SWU are written linearly, similar to spoken languages. Figure 3 gives an example of the relationship between SignWriting, FSW, and SWU[6]. We use FSW in our research instead of SWU to explore the potential of factorizing SignWriting symbols and utilizing numerical values of their position (§3.3, §3.4).

---

[4]In the paper, we distinguish between a phonetic "writing system" (e.g., SignWriting) and "glosses" (lexical notation, marking the semantics of each sign with a distinct category).

[5]Related work based on HamNoSys: Morrissey (2011); Sanaullah et al. (2021); Walsh et al. (2022)

[6]Online demonstration: https://slevinski.github.io/SuttonSignWriting/characters/index.html.

```
AS14c20S27106M518x529S14c20481x471S27106503x489
AS18701S1870aS2e734S20500M518x533S1870a489x515S18701482x490S20500508x496S2e734
500x468 S38800464x496
```
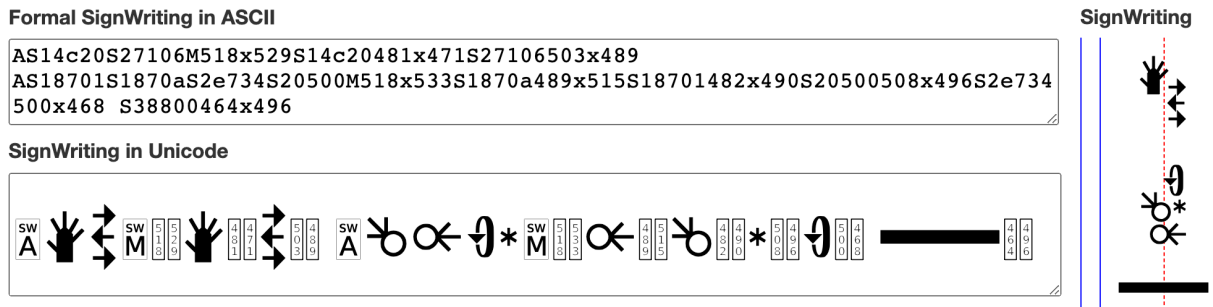
SignWriting in Unicode

Figure 3: "Hello world." in FSW, SWU and SignWriting graphics. In FSW/SWU, A/SWA and M/SWM are the box markers (acting as sign boundaries); S14c20 and S27106 (graphemes in SWU) are the symbols; 518 and 529 are the x, y positional numbers on a 2-dimensional plane that denote symbols' position within a sign box, S38800 (horizontal bold line in SWU) is the punctuation full stop symbol.

3 Data and method

The data source we use for this research is Sign-Bank, the largest repository of *SignPuddles*[7]. A SignPuddle is a community-driven dictionary where users add parallel examples of SignWriting and spoken language text (not necessarily with corresponding videos and glosses). The puddles contain material from various signed languages and linguistic domains (e.g., general literature or Bible) without a strict writing standard. We use the Sign Language Datasets (Moryossef and Müller, 2021) library to load SignBank as a Tensorflow Dataset.

3.1 Data statistics

In SignBank, there are roughly 220k parallel samples from 141 puddles covering 76 language pairs, yet the distribution is unbalanced (full details in Appendix C). Relatively high-resource language pairs (over 10k samples) are listed in Table 1.

Notably, most of the puddles are dictionaries, which we consider less valuable than sentence pairs (instances of continuous signing) for a general MT system. If dictionaries are used as training data, we expect models to memorize word mappings and not learn to generate sentences.

Therefore, we treat the four sentence-pair puddles (Table 2) of the relatively high-resource language pairs as primary data and the other dictionary puddles as auxiliary data. Note that even the language pairs constituting the high-resource pairs of SignBank are low-resource compared to datasets used in mature MT systems for spoken languages, where millions of parallel sentences are commonplace (Akhbardeh et al., 2021).

3.2 Data preprocessing

We first perform general data cleaning to extract the main body of spoken language text and remove irrelevant parts such as HTML tags or samples that are empty or too long (100 words for a dictionary entry). We then learn a byte pair encoding (BPE) segmentation (Sennrich et al., 2016) on the cleaned spoken language text, using a vocabulary size of 2,000.

Multilingual models In our multilingual experiments (§4.2, §4.3), we learn a shared BPE model across all spoken languages.

Following Johnson et al. (2017), we add special tags at the beginning of source sequences to indicate the desired target language and nature of the training data (sentence pair or dictionary). Three types of tags are designed to encode all necessary information: (a) spoken language code; (b) country code[8]; (c) dictionary vs. sentence pair. For example, an English sentence to be translated into American Sign Language is represented as the following:

```
<2en> <4us> <sent> Hello world.
```

Data split We shuffle the data and split it into 95%, 3%, and 2% for training, validation, and test sets, respectively.

3.3 FSW parsing

On the sign language side, an appropriate segmentation and tokenization strategy is needed for the FSW data. We parse an original FSW sequence (e.g. Figure 3) into several pieces:

- box markers: A, M, L, R, B;

---

[8]spoken language code plus country code specifies a one-to-one mapping to a related signed language in our data.

language pair	#samples	#puddles
en-us (American English & American Sign Language)	43,698	7
pt-br (Brazilian Portuguese & Brazilian Sign Language)	42,454	3
de-de (Standard German & German Sign Language)	24,704	3
fr-ca (Canadian French & Quebec Sign Language)	11,189	3

Table 1: Relatively high-resource language pair statistics.

puddle name	language pair	#samples	#signs	mean sequence len
Literature US	en-us	700	9,922	24
ASL Bible Books NLT	en-us	11,667	51,485	24
ASL Bible Books Shores Deaf Church	en-us	4,321	44,612	31
Literatura Brasil	pt-br	1,884	19,221	13

Table 2: Primary sentence-pair puddles. Mean sequence length is measured by the mean number of words in the spoken language sentences.

- symbols: S1f010, S18720, etc.;

- positional numbers x and y: 515, 483, etc.;

- punctuation marks (special symbols without box markers): S38800, etc.

We further factorize each symbol into several parts regarding its orientation (see Figure 7 in Appendix A for an explicit motivation of this step). For example, the symbol S1f010 is split into:

- symbol core: S1f0;

- column number (from 0 to 5): 1;

- row number (from 0 to hex F): 0.

For positional numbers, which have a large range (from 250 to 750) and are encoded discretely, we hypothesise that models might have difficulty understanding their relative order. Therefore, we further calculate two additional factors that denote a symbol's relative position (based on the absolute numbers) within a sign: relative x and relative y, both ranging from 0 to #symbols - 1.

We provide a full example of the result of FSW parsing in Listing 1 in Appendix C.

3.4 Factored machine translation

We use a factored machine translation system (Koehn and Hoang, 2007; Garcia-Martinez et al., 2016) to encode or decode parsed FSW sequences. We argue that this architecture is suitable because

	Source	Target
symbol	S1f010	
X	515	
Y	483	
relative X	0	→ "Hi"
relative Y	1	
symbol core	S1f0	
column	1	
row	0	

Figure 4: Representation of translating a FSW symbol together with its factors to English.

concatenating all parsed FSW tokens results in sequences much longer than the maximum length of many Transformer models (e.g., 512).

From another perspective, the essential information units are the symbols. Nevertheless, the positional numbers are necessary to determine how symbols are assembled. The same symbols can be arranged differently in space to convey different meanings.

In our setup, we treat the symbols (including punctuation marks and box markers) as the primary source/target tokens and the rest as source/target factors that are strictly aligned with each source/target token (illustrated by Figure 4).

Depending on the translation direction, factored FSW representations need to be encoded or decoded. For encoding (when FSW is the source), we embed each factor separately and then concatenate

them to the aligned symbol's embedding. For decoding (when FSW is the target), we use only a subset of factors (absolute x and y) because others are irrelevant for prediction, and additional weighted cross-entropy losses are calculated.

4 Experiments and results

This section introduces three lines of experiments on both bilingual and multilingual SignWriting translation. We use Transformer models (Vaswani et al., 2017) that support source and target factors. See Appendix B for more details on our training configuration.

4.1 Initial exploration with a bilingual model

For a first exploration, we train a bilingual model that translates from American Sign Language (ASL) to English (en-us). The purpose of this experiment is (a) to demonstrate that automatic SignWriting translation is feasible and (b) to explore different strategies for data processing and hyperparameters.

We use roughly 40k parallel training samples comprising roughly 15k sentence pairs and 25k dictionary pairs. The quality of spoken language translation is measured by BLEU (Papineni et al., 2002) and chrF2 (Popović, 2015). Table 3 shows the evaluation results on the test set.

4.2 Multilingual sign-to-spoken translation

Here we extend our initial bilingual model to a multilingual setting, translating from multiple signed languages to multiple spoken languages. We define two data conditions:

- **high-resource:** using roughly 100k parallel training samples (roughly 17k sentence pairs and roughly 83k dictionary pairs covering four language pairs),

- **adding low-resource:** in addition to the high-resource data, use all additional language pairs in SignBank that have at least 1k parallel samples (most of which are dictionaries). The total number of training examples grows to roughly 170k, covering 21 language pairs (Table 7).

The exact factorization strategy and model hyperparameters are informed by our bilingual experiments reported in Table 3.

Evaluating dictionary entries For these multilingual models, many of the training samples are dictionary entries, and so are some test samples. To evaluate the translation quality for dictionary entries, we use top-n accuracy, which tests whether one of the top-n translation candidates from beam search matches the entry from the reference.

Table 4 shows the evaluation results on the test set.

4.3 Multilingual spoken-to-sign translation

Finally, we train multilingual models that translate in the reverse direction, from spoken languages to signed languages. The data and model configuration are the same as for the multilingual sign-to-spoken model under high-resource data condition.

FSW decoding strategies SignWriting utterances are parsed into a factored FSW representation (§3.3, §3.4) and are used for encoding successfully, yet it is not obvious how to best decode to FSW. We try the following strategies: (a) predicting everything (including positional numbers) as target tokens all in one long target sequence, inspired by Chen et al. (2022); (b) predicting symbols only (as a comparative experiment); (c) predicting symbols with positional numbers as target factors.

During decoding in the test phase, we apply beam search only for the main target token prediction. Target factors do not participate in beam search, i.e., each target factor prediction is the argmax of the corresponding output layer distribution. We shift target factors to the right by 1 to condition their prediction on the previously generated target symbol.

Evaluation of FSW output Due to variations of SignWriting symbols based on different orientations that do not change meaning (Figure 7), evaluating FSW output only at the token (symbol) level is not sufficient. Therefore, we evaluate the output symbols (e.g., line 4 in Listing 1) not only with BLEU, but also chrF2++, which captures both word-level and character-level statistics. Additionally, we evaluate the output positional numbers by mean absolute error (MAE) to measure the distance between predicted positional numbers and the ones from the FSW reference (e.g., lines 5 and 6 in Listing 1). Let x be the predicted sequence of positional numbers and y be the gold sequence:

$$MAE(x,y) = \frac{1}{|x|} \sum_{i=1}^{|x|} |x_i - y_i|$$

	model	BLEU	chrF2
E1	baseline (lowercase training and test data)	22.5	-
E2	E1 + dictionary data	25.2	-
E3	E2 + BPE	27.0	46.2
E4	E3 + x,y as factors	27.5	46.5
E5	E4 + symbol core as source + row, col as factors	23.1	41.2
E6	E4 + relative x, y as factors	28.1	47.5
E7	E6 + aggressive dropout + tied softmax	31.4	52.0
E8	E7 + symbol core, row, column as factors	32.0	52.7
E9	E8 + remove lowercasing	30.8	51.2
E10	E9 + smaller BPE vocab 2000 to 1000	29.5	50.8

Table 3: Translation quality of ASL→en-us bilingual models. Note that E1 to E8 are trained and evaluated with all spoken language data lowercased, while from E9 to all later experiments we remove the lowercasing, so we expect a little performance drop for the later experiments. We introduce chrF2 as an evaluation metric starting from E3.

language	metrics	4 language pairs (100k)	21 language pairs (170k)
en-us (40k)	BLEU	29.5	25.0
	chrF2	49.8	47.0
	top-1	0.37	0.33
	top-5	0.52	0.45
en-sg (1k)	top-1	-	0.20
	top-5	-	0.27
pt-br (40k)	BLEU	23.8	6.4
	chrF2	44.3	17.5
	top-1	0.12	0.09
	top-5	0.17	0.15
mt-mt (4k)	BLEU	-	10.1
	chrF2	-	29.8
	top-1	-	0.05
	top-5	-	0.05
de-de (20k)	top-1	0.22	0.15
	top-5	0.31	0.27
de-ch (4k)	top-1	-	0.04
	top-5	-	0.06
fr-ca (10k)	top-1	0.04	0.07
	top-5	0.08	0.10
fr-fr (1k)	top-1	-	0.16
	top-5	-	0.24
fr-ch (8k)	top-1	-	0.07
	top-5	-	0.09

Table 4: Translation quality of multilingual sign-to-spoken models (partial results on the most frequent languages). Languages without sentence pairs are only evaluated by top-n accuracy. Empty cells mean that a language pair is not supported by the model. In the parentheses are the rough numbers of samples.

	model	BLEU	chrF2++	MAE x	MAE y
E1	2symbol+numbers	6.6	23.1	-	-
E2	2symbol	25.6	44.2	-	-
E3	2symbol+factors (w=1)	19.9	39.1	46.5	52.6
E4	2symbol+factors (w=0.5)	21.9	40.8	46.8	52.7
E5	2symbol+factors (w=0.2)	22.9	42.0	47.4	53.0
E6	2symbol+factors (w=0.1)	22.0	41.7	46.4	52.2
E7	2symbol+factors (w=0.01)	21.0	40.9	48.4	58.3

Table 5: Translation quality of multilingual spoken-to-sign models. Evaluated in BLEU (on symbol, higher is better), chrF2++ (on symbol, higher is better), and MAE (on positional numbers, lower is better). w denotes the weight between each factor's loss and the main target loss.

language	BLEU (on symbols)	chrF2++ (on symbols)
en-us (40k)	35.7	58.4
pt-br (40k)	1.9	14.9
de-de (20k)	17.3	43.2
fr-ca (10k)	5.3	19.1

Table 6: Translation quality of multilingual spoken-to-sign model (w=0.1) per language. In the parentheses are the rough numbers of samples per language.

where if the predicted and gold sequences do not have the same length, they are padded with zeros.

Table 5 shows the results of evaluation on the test set. Table 6 shows the results of multilingual evaluation on *E6* (w=0.1) of Table 5.

5 Discussion

5.1 Effect of adding dictionaries, BPE, and low-resource optimizations

As shown in Table 3, enlarging the sentence-level training data (15k sentence pairs) with 25k dictionary pairs improves the translation quality by 2.7 BLEU (*E1* vs. *E2*). Likewise, applying BPE segmentation to the spoken language side also improves translations by 1.8 BLEU (*E2* vs. *E3*).

We evaluate several low-resource "tricks" (Sennrich and Zhang, 2019) including aggressive dropout and weight tying[9]. These low-resource optimizations borrowed from spoken language MT prove to be effective for sign language translation as well, as they result in an improvement of 3.3 BLEU and 4.5 chrF2 (*E6* vs. *E7* in Table 3).

---

[9]The tying is only between the target embedding and the softmax output matrix since the source and target languages are of a very different nature and therefore cannot be tied.

5.2 Utilizing positional numbers

In earlier sections we introduce novel methods to parse and factorize FSW (§3.3, §3.4). However, from a model training perspective it is unclear how to best utilize additional factors such as positional numbers. In *E4*, *E5*, *E6*, and *E8* of Table 3, we explore different ways of including factors.

We find that the best strategy is explicitly adding all additional information (x, y, relative x, relative y, symbol core, column number, row number) as source factors while keeping symbols as the primary source tokens. This strategy achieves the state-of-the-art performance of 32.0 BLEU and 52.7 chrF2 in *E8*.

5.3 Generating positional numbers

We explore different ways of generating positional numbers in Table 5. As a first attempt, we treat them as normal target tokens in *E1*, which results in poor performance and overly long target sequences, and long beam search decoding time.

In all subsequent experiments, we treat positional numbers as target factors and generally achieve over 20 BLEU (evaluating on symbols). In *E2*, we translate only the symbols as a baseline. Then we also try translating with target factors and varying the weights between factors and the pri-

mary target, i.e., symbols. Finally, we observe that *E6* (w=0.1) leads to the best trade-off between generating symbols and positional numbers.

5.4 Multilingual performance

We discuss multilingual performance mainly based on Table 4. Generally speaking, the more resources a language has in the multilingual model, the better its performance (Zhou et al., 2021). The two target languages most frequent in the training data–American English (en-us, 40k) and Brazilian Portuguese (pt-br, 40k)–have the highest translation quality.

Multilingual transfer effects   We observe examples of both positive and negative multilingual transfer. Evidence shows that a relatively high-resource language can help a related low-resource language. For instance, the performance of Singaporean English (en-sg, 1k) is likely improved by American English (en-us, 40k), which is almost as good as Standard German (de-de, 20k).

The comparable bilingual en-us model (*E9* in Table 3) outperforms en-us in our multilingual sign-to-spoken model in Table 4 by 1.3 BLEU and 1.4 chrF2. However, when extending the training data from 4 to 21 language pairs, we observe severe degradations: en-us drops by 4.5 BLEU and 2.8 chrF2; Brazilian Portuguese (pt-br) drops by 17.4 BLEU and 26.8 chrF2.

Such findings are in line with previous work on highly multilingual translation systems. For example, Aharoni et al. (2019) finds that average per-language performance drops when the number of languages increases. We conclude that Sign-Writing translation suffers from a similar *curse of multilinguality*.

5.5 Side-by-side SignWriting example

Finally, to gain intuition for how well the translation model signs, we give a side-by-side example of SignWriting graphics. We compare the reference and model prediction of an ASL utterance corresponding to an American English utterance from the Bible corpus, shown in Figure 5.

We ask an ASL user proficient in SignWriting for a translation of the predicted SignWriting back to English to assess the quality of the prediction.

Similar patterns appear in both: in the beginning, the model signs "Verse 41" in the same way as in the reference; the graphics in the top parts of all the columns are consistent; and we see correct sym-
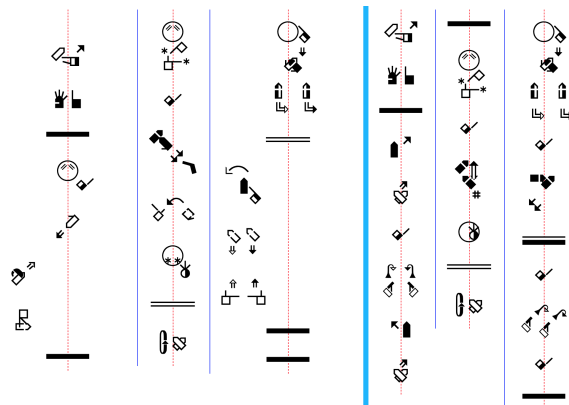

Figure 5: Side-by-side SignWriting example. ASL translation of the English sentence "Verse 41. He gave her his hand and helped her up. Then he called in the widows and all the believers, and he presented her to them alive." Separated by the vertical bold (light blue) line, the left is the gold sentence, and the predicted sentence is on the right. The predicted sentence translated back to English is "Verse 41. His hand he gives her hand. Then he helped up, all believers he warned: he put there."

bols sometimes predicted with slightly different positions.

More translation examples can be seen in Appendix D.

6 Conclusion

This work explores building bilingual and multilingual translation systems between spoken and signed languages. Instead of representing sign language as videos (or as continuous features derived from videos) common in previous research, we propose to represent sign language in SignWriting, a sign language writing system. We argue that using a written form is more amenable to well-established NLP techniques.

However, encoding or decoding SignWriting in an MT system requires specialized tools. Therefore, we propose novel methods to parse, factorize, decode, and evaluate SignWriting sequences. Our factorization technique divides SignWriting sequences into meaningful units such as sign symbols and positional numbers. The factors are then encoded or decoded by a factored Transformer model.

As a result, we achieve over 30 BLEU in the bilingual setting and over 20 BLEU for some high-resource language pairs in both directions in the multilingual setting.

Using SignWriting as an intermediate representation enables us to reuse tools (e.g., evaluation met-

rics) from spoken language translation. We also observe striking similarities to spoken language MT in the experiments themselves. For example, low-resource optimizations have a similar impact, and multilingual models exhibit similar transfer effects. These findings validate our use of an intermediate text representation for signed languages to include them in NLP research.

7 Limitations

7.1 A word on top-n accuracy

In the translation of dictionary data, if a dictionary entry has been seen during training, assuming the model has enough capacity, it should memorize and predict it. However, evaluating the translation is tricky, so in §4.2 we resort to using top-n accuracy.

Paradoxically, high top-n accuracy on the test set does not guarantee good generalization and might be associated with overlap between the training and test sets. Conversely, claiming a model is terrible when it performs poorly on the test set is unjustified, as there might be no overlap between the training and test sets. If a model has seen all the words from a language, it should perform well on whatever dictionary test set. However, this is not the case in our low-resource setting.

7.2 Fingerspelling tokenization

Fingerspelling (Battison, 1978; Wilcox, 1992; Brentari and Padden, 2001) is an interesting linguistic phenomenon where a signed language geographically coexists with a spoken language. For words with no associated signs (e.g., names of people, locations, organizations, etc.), sign language users borrow a word of a spoken language by spelling it letter-by-letter with predefined signs for the letters of the alphabet of that language. The fingerspelling (manual) alphabet of a sign language draws on a closed set of hand shapes, which are supported by SignWriting.

As fingerspelling is usually applied on a character level (rarely extending to the level of multiple characters, such as "CH" or "SCH" for the finger alphabet of Swiss German Sign Language), the way BPE segmentation works (on subword level) does not apply perfectly. However, if we could detect fingerspelling during the segmentation/tokenization process, then force fingerspelled words to be split letter-by-letter, our models should be able to learn better the mappings between fingerspelling signs and spoken language letters.

7.3 Towards better multilingual models

As shown by Table 2, the data we use to train our models only contains many sentence pairs for American English and American Sign Language. For other language pairs, we train mainly on dictionary data.

At the time of writing, we find a multilingual parallel corpus created from translations of the Bible[10] (Christodoulopoulos and Steedman, 2014), which, if aligned correctly, can be used to translate the ~15k American Sign Language biblical text to another 100+ spoken languages. We believe we could train better multilingual translation models (at least on the spoken language side) based on them.

7.4 Regression objective for positional numbers

In our experiments, positional numbers are treated as target factors (§5.3), contributing cross-entropy loss to the training process. However, we are aware that the positional numbers are, by nature, numeric values, so a regression objective/loss would possibly work better than the current cross-entropy loss, as it better reflects the numeric relationship between positional numeric values.

As for now, the target factor function we use is only implemented with a classification objective (cross-entropy loss). We envision that custom implementation of the regression objective might improve translation quality in this scenario.

7.5 Possibly flawed positional number evaluation

We note that using MAE for evaluating positional numbers (§4.3) is possibly flawed because the predicted symbol sequences can deviate from the gold symbol sequences. If this is the case, making a token-by-token comparison on the positional numbers is meaningless, as even the sequence length can mismatch.

7.6 Advanced SignWriting evaluation

Finally, we call for advanced and novel methods of SignWriting evaluation, considering its differences from spoken languages.

In our experiments, we separate the evaluation of FSW symbols and positional numbers. For symbols, we borrow BLEU and chrF2/chrF2++ from spoken language evaluation since FSW symbols are the basic graphemes in SignWriting that show

---

[10]https://github.com/christos-c/bible-corpus

many similar linguistic features as spoken language words. For positional numbers, MAE is used, and its limitation is discussed in §7.5.

From a broader perspective, FSW is merely a linearized specification of SignWriting, which means we can also evaluate on the original graphical form, as we do manually in §5.5. Moreover, we can exploit CV techniques to do an automatic comparative evaluation between predicted SignWriting graphics and gold SignWriting graphics.

Ideally, a cascading evaluation method is applied to SignWriting: we first evaluate the overall graphics of the signs, then the symbols within the signs, then the position of the symbols, then the factorized representation of the symbols. Finally, a thorough human evaluation is needed to gain better insight.

Note on reproducibility

We will release the source code and documentation to train our models, an API server with the trained models, and a demo Web application. This will allow others to see and consistently reproduce our results with minimal changes. We encourage the community to attempt to reproduce our results and publish the results.

Acknowledgements

References

Nikolaos M. Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George Xydopoulos, Klimis Antzakas, Dimitris Papazachariou, and Petros Daras. 2021. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, page 1–1.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual 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)*, pages 3874–3884.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Robbin Battison. 1978. *Lexical borrowing in American sign language.* ERIC, Linstok Press, Inc., Silver Spring, Maryland 20901.

Mark Borg and Kenneth P Camilleri. 2019. Sign language detection "in the wild" with recurrent neural networks. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1637–1641. IEEE.

Yosra Bouzid and Mohamed Jemni. 2013. An avatar based approach for automatically interpreting a sign language notation. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 92–94.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. 2019. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, page 16–31. Association for Computing Machinery.

Diane Brentari and Carol Padden. 2001. A language with multiple origins: Native and foreign vocabulary in american sign language. *Foreign vocabulary in sign language: A cross-linguistic investigation of word formation*, pages 87–119.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort. 2020. Automatic segmentation of sign language into subtitle-units. In *European Conference on Computer Vision*, pages 186–198. Springer.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020a. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020b. 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*, pages 10023–10033.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2021. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):172–186.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. 2022. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*.

Christos Christodoulopoulos and Mark Steedman. 2014. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49:1–21.

Iva Farag and Heike Brock. 2019. Learning motion disfluencies for automatic sign language segmentation. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7360–7364. IEEE.

Mercedes Garcia-Martinez, Loïc Barrault, and Fethi Bougares. 2016. Factored Neural Machine Translation Architectures. In *International Workshop on Spoken Language Translation (IWSLT'16)*.

Binyam Gebrekidan Gebre, Peter Wittenburg, and Tom Heskes. 2013. Automatic sign language identification. In *2013 IEEE International Conference on Image Processing*, pages 2626–2630. IEEE.

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458. European Association for Machine Translation.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Trevor Johnston and Louise De Beuzeville. 2016. Auslan corpus annotation guidelines. *Auslan Corpus*.

Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876. Association for Computational Linguistics.

Reiner Konrad, Thomas Hanke, Gabriele Langer, Susanne König, Lutz König, Rie Nishio, and Anja Regen. 2018. Public dgs corpus: Annotation conventions. Technical report, Hamburg University.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. 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): System Demonstrations*, pages 109–114. Association for Computational Linguistics.

Johanna Mesch and Lars Wallin. 2012. From meaning to signs and back: Lexicography and the swedish sign language corpus. In *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon [Language Resources and Evaluation Conference (LREC)]*, pages 123–126.

Caio DD Monteiro, Christy Maria Mathew, Ricardo Gutierrez-Osuna, and Frank Shipman. 2016. Detecting and identifying sign languages through visual features. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 287–290. IEEE.

Sara Morrissey. 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.

Amit Moryossef and Yoav Goldberg. 2021. Sign Language Processing. https://sign-language-processing.github.io/.

Amit Moryossef and Mathias Müller. 2021. Sign language datasets. https://github.com/sign-language-processing/datasets.

Amit Moryossef, Ioannis Tsochantaridis, Roee Yosef Aharoni, Sarah Ebling, and Srini Narayanan. 2020. Real-time sign-language detection using human

pose estimation. In *SLRTP 2020: The Sign Language Recognition, Translation & Production Workshop*.

Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022a. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mathias Müller, Zifan Jiang, Amit Moryossef, Annette Rios, and Sarah Ebling. 2022b. Considerations for meaningful sign language machine translation based on glosses. *arXiv preprint arXiv:2211.15464*.

Achraf Othman and Mohamed Jemni. 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*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. Association for Computational Linguistics.

Siegmund Prillwitz and Heiko Zienert. 1990. Hamburg notation system for sign language: Development of a sign writing with computer application. In *Current trends in European Sign Language Research. Proceedings of the 3rd European Congress on Sign Language Research*, pages 355–379.

Muhammad Sanaullah, Babar Ahmad, Muhammad Kashif, Tauqeer Safdar, Mehdi Hassan, Mohd Hilmi Hasan, and NorShakirah Aziz. 2021. A real-time automatic translation of text to sign language. *Computers, Materials and Continua*, 70:2471–2488.

Pinar Santemiz, Oya Aran, Murat Saraclar, and Lale Akarun. 2009. Automatic sign segmentation from continuous signing via multiple sequence alignment. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2001–2008. IEEE.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020a. Adversarial training for multi-channel sign language production. In *The 31st British Machine Vision Virtual Conference*. British Machine Vision Association.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020b. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020c. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pages 687–705.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221.

Steve Slevinski. 2021. Formal SignWriting. Internet-Draft draft-slevinski-formal-signwriting-08, Internet Engineering Task Force.

D. Stiehl, L. Addams, L. S. Oliveira, C. Guimarães, and A. S. Britto. 2015. Towards a signwriting recognition system. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 26–30.

Valerie Sutton. 1990. *Lessons in sign writing*. Sign-Writing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Harry Walsh, Ben Saunders, and Richard Bowden. 2022. Changing the representation: Examining language representation for neural sign language production. In *LREC 2022*.

Sherman Wilcox. 1992. *The phonetics of fingerspelling*, volume 4. John Benjamins Publishing.

Qinkun Xiao, Minying Qin, and Yuting Yin. 2020. Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks*, 125:41–55.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proceedings of the 59th Annual Meeting of the*

Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7347–7360, Online. Association for Computational Linguistics.

Kayo Yin and Jesse Read. 2020. Better sign language translation with stmc-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989.

Jan Zelinka and Jakub Kanis. 2020. Neural sign language synthesis: Words are our glosses. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3395–3403.

Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. 2000. A machine translation system from english to american sign language. In *Conference of the Association for Machine Translation in the Americas*, pages 54–67. Springer.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5664–5674. Association for Computational Linguistics.

A   Extended introduction to SignWriting

SignWriting (Sutton, 1990) is a sign language writing system developed by Valerie Sutton[11] and currently managed by Steve Slevinski[12]. SignWriting is very featural and visually iconic, both in:

- the shapes of the symbols, which are abstract pictures of hand shapes (Figure 6), orientation (Figure 7), body locations, facial expressions, contacts, and movement;

- the symbols' two-dimensional spatial arrangement in an invisible "sign box" (Figure 8).

Outside each sign, the script is written linearly to reflect the temporal order of signs. Signs are mostly written vertically, arranged from top to bottom within each column, interspersed with special punctuation symbols (horizontal lines), and the columns progress left to right across the page. Within each column, signs may be vertically aligned to the center or shifted left or right to indicate body shifts.

A.1   Formal SignWriting in ASCII (FSW)

In 2012, Formal SignWriting in ASCII (FSW) specification (Slevinski, 2021) was released and documented in an Internet Draft submitted to the IETF.

The design of FSW is computerized so that it can be recognized and processed by programs. While signed languages are natural languages, FSW is a formal language handy in mathematics, computer science, and linguistics.

Although SignWriting is two-dimensional, FSW is written linearly like spoken languages. Each sign is written as first a box marker, then a sequence of symbols, and their relative position, as illustrated by Figure 3.

A.2   SignWriting in Unicode (SWU)

In 2017, SignWriting in Unicode (SWU) specification (Slevinski, 2021) was released, making SignWriting included in the Unicode Standard. The Unicode block for SWU is U+1D800 - U+1DAAF.

As illustrated in Figure 3, SWU is also written linearly. FSW and SWU are isomorphic and interchangeable, and both faithfully encode the complete information of SignWriting.
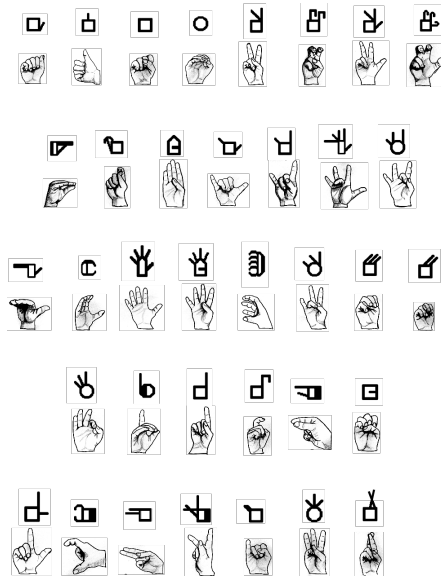
---

[11]Valerie Sutton: https://en.wikipedia.org/wiki/Valerie_Sutton

[12]Steve Slevinski: https://steveslevinski.me/

Figure 6: Hand shapes and their equivalents in SignWriting.

Figure 7: Orientation of a symbol in SignWriting in 3D space. Each row applies a rotation of the palm in a 2D space **vertical** to the ground. Each column applies a rotation of the palm in a 2D space **parallel** to the ground. This can be seen as a factorization of the symbol S100xx to its core S100 plus row and column numbers.
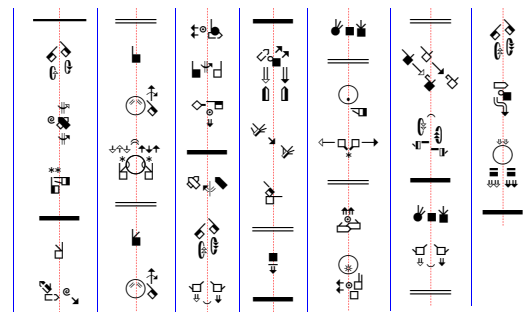

Figure 8: An example of SignWriting written in columns, ASL translation of an introduction to Formal SignWriting in ASCII. The relative positions of the symbols within the box iconically represent the locations of the hands and other parts of the body involved in the sign being represented.

1719

B Experimental setup

We performed all our experiments with Python 3.8.11 on an Nvidia Tesla V100 GPU (32GB GPU ram).

B.1 40k sign-to-en-us

Data includes:

- ∼15k sentence pairs from 3 en-us puddles: Literature US, ASL Bible Books NLT, ASL Bible Books Shores Deaf Church,

- ∼25k dictionary pairs from 3 en-us puddles: Dictionary US, LLCN & SignTyp, ASL Bible Dictionary,

which leads to:

- ∼6k source vocabulary size (number of non-factorized symbols),

- ∼2k target vocabulary size (determined by BPE).

Final model configuration:

- 6 layers + 8 heads + 512 embedding size (16 for each factor) (0.5 dropout) + 512 hidden size (0.5 dropout) + 2,048 feed forward size (0.5 dropout),

- initial learning rate 0.0001, decrease learning rate by a factor of 0.7 every 5 times validation score (BLEU) not improved,

- batch size 32 sentences, label smoothing 0.2, epochs 300,

- for testing, decoding with a checkpoint with the best validation score, beam size 5, alpha for length penalty 1.

Experiments were conducted with a custom version of Joey NMT (Kreutzer et al., 2019) to support source factors. Each model (∼47 million parameters) finished training within 1 day.

B.2 100k sign-to-spoken

Data includes:

- ∼17k sentence pairs from 3 en-us puddles (Literature US, ASL Bible Books NLT, ASL Bible Books Shores Deaf Church) and 1 pt-br puddle (Literatura Brasil),

- ∼83k dictionary pairs from 3 en-us puddles (Dictionary US, LLCN & SignTyp, ASL Bible Dictionary), 2 pt-br puddles (Dicionário Brasil, Enciclopédia Brasil), 1 de-de puddle (Wörterbuch DE) and 1 fr-ca puddle (Dictionnaire Quebec),

which leads to:

- ∼11k source vocabulary size (number of non-factorized symbols),

- ∼2k target vocabulary size (determined by BPE).

A little change to the previous configuration to make training more efficient:

- batch size 4,096 tokens.

Experiments were conducted with a custom version of Joey NMT to support source factors. Each model (∼50 million parameters) finished training within ∼1.5 days and ∼3 days, respectively.

B.3 100k spoken-to-sign

Data and model configurations are the same as **100k sign-to-spoken**, except that we use perplexity (Chen and Goodman, 1996; Sennrich, 2012) as validation score instead of BLEU.

Experiments were conducted with Sockeye (Hieber et al., 2020) for the convenience of ready-to-use target factor support. Each model (∼60 million parameters) finished training within ∼0.5 day.

C Data

Figure 9 visualizes the language pair distribution in SignBank. Table 7 contains an exhaustive list of all 21 language pairs used in this research. Listing 1 shows an example of FSW parsing and factorization.
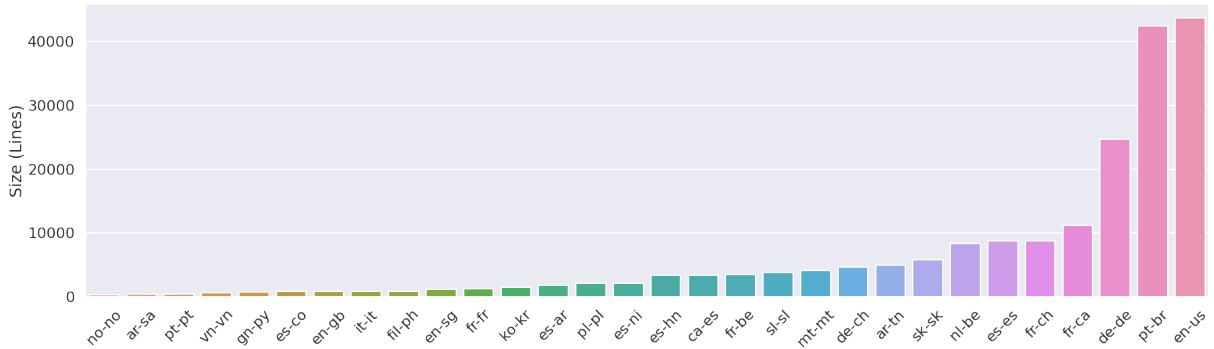

Figure 9: Data distribution (the first 30 language pairs).

language	#samples	#puddles	sentence pairs (>1k)
en-us (American English)	43,698	7	✓
en-sg (Singaporean English)	1,136	2	
pt-br (Brazilian Portuguese)	42,454	3	✓
mt-mt (Maltese Maltese)	4,118	4	✓
de-de (German German)	24,704	3	
de-ch (Swiss German)	4,700	2	
fr-ca (Canadian French)	11,189	3	
fr-ch (Swiss French)	8,806	3	
fr-be (Belgian French)	3,439	1	
fr-fr (French French)	1,299	2	
es-es (Spanish Spanish)	8,806	2	
es-hn (Honduran Spanish)	3,399	1	
es-ni (Nicaraguan Spanish)	2,150	2	
es-ar (Argentinian Spanish)	1,774	2	
ar-tn (Tunisien Arabic)	4,965	2	
ca-es (Spanish Catalan)	3,419	2	
ko-kr (Korean Korean)	1,525	1	
nl-be (Belgian Flemish)	8,301	2	
pl-pl (Polish Polish)	2,130	2	
sk-sk (Czech Czech)	5,780	2	
sl-sl (Slovenian Slovenian)	3,808	2	

Table 7: All 21 language pairs (spoken languages with corresponding signed languages).

```
1  {
2    'fsw': 'M550x535S32a00482x483S15d09455x499S15d01522x497S22114516x484
3      S22114456x484S20f00524x522S20f00451x523',
4    'symbol': 'M S32a00 S15d09 S15d01 S22114 S22114 S20f00 S20f00',
5    'feat_x': '550 482 455 522 516 456 524 451',
6    'feat_y': '535 483 499 497 484 484 522 523',
7    'feat_x_rel': '-1 3 1 5 4 2 6 0',
8    'feat_y_rel': '-1 0 4 3 1 2 5 6',
9    'feat_core': 'M S32a S15d S15d S221 S221 S20f S20f',
10   'feat_col': '-1 0 0 0 1 1 0 0',
11   'feat_row': '-1 0 9 1 4 4 0 0',
12 }
```

Listing 1: An example of FSW parsing and factorization.

D   More side-by-side SignWriting examples

Separated by the vertical bold (light blue) line, the left is the gold sentence, and the predicted sentence is on the right.
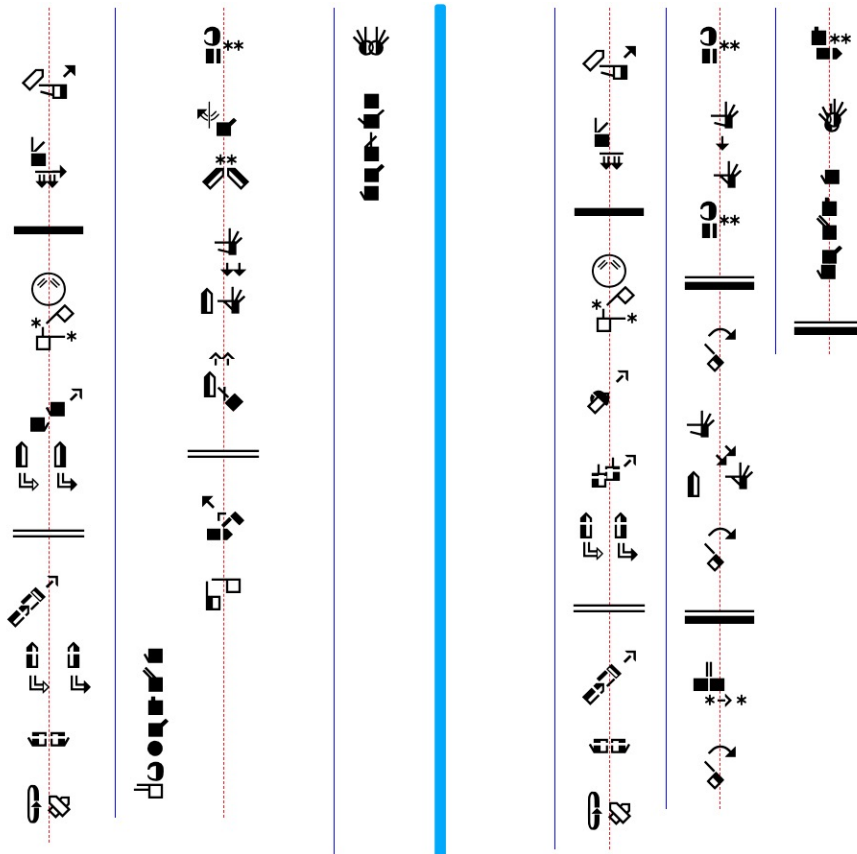

Figure 10: ASL translation of the English sentence "Verse 22. Then the apostles and elders together with the whole church in Jerusalem chose delegates, and they sent them to Antioch of Syria"
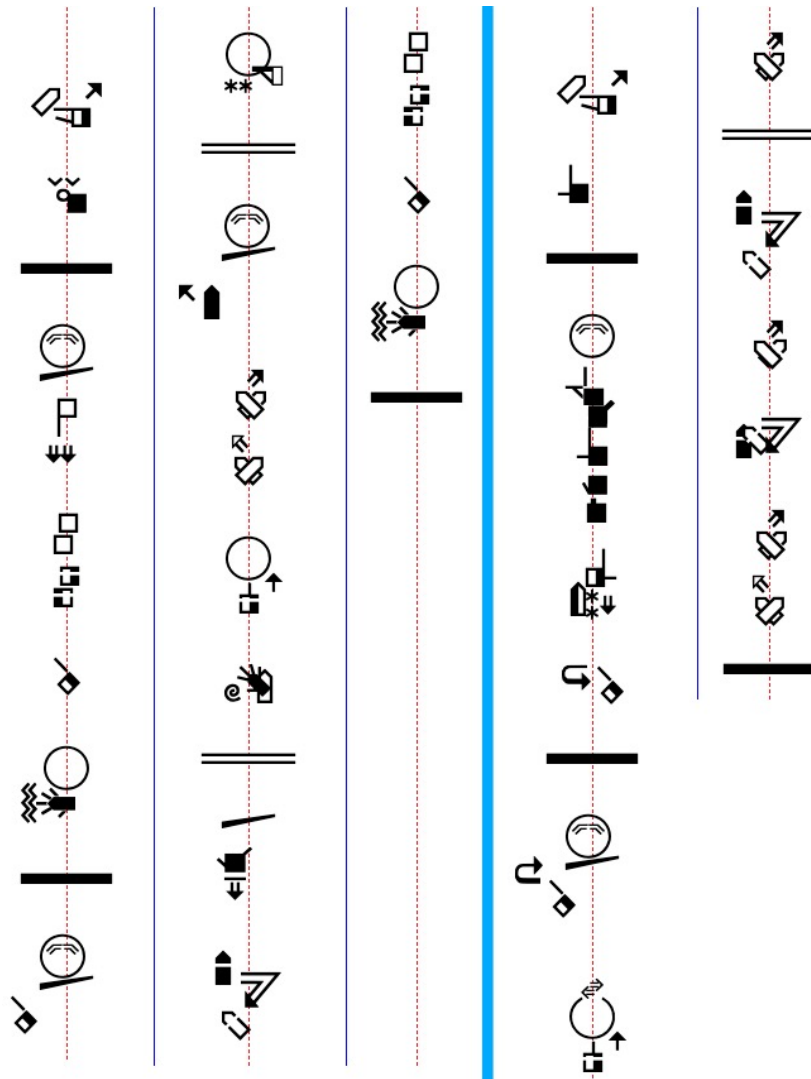
Figure 11: ASL translation of the English sentence "These are what defile you. Eating with unwashed hands will never defile you."
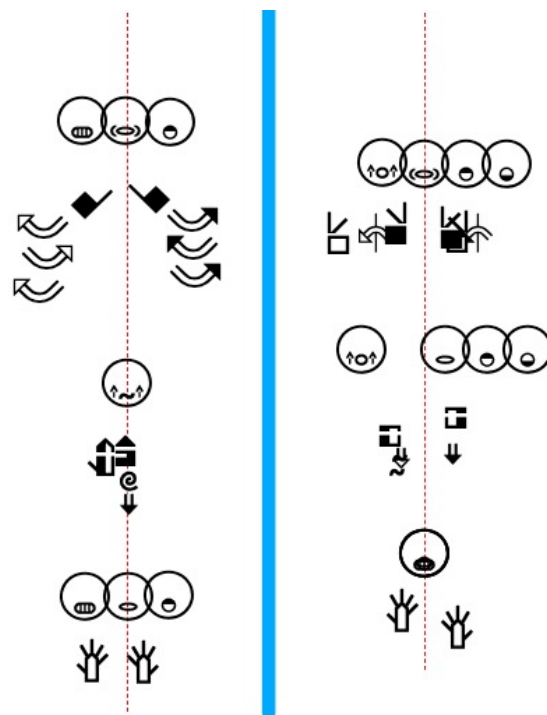
Figure 12: German Sign Language translation of the German words "signen, senken, zehn"