Universität Zürich[UZH]

Master's thesis

for obtaining the academic degree

Master of Science

in Department of Informatics

Machine Translation between Spoken Languages and Signed Languages in Written Form

Author: Zifan Jiang

Student ID number: 19-757-467

Advisors:

Dr. Mathias Müller

M.Sc. Amit Moryossef (Bar-Ilan University, Tel-Aviv, Israel)

Supervisor: Prof. Dr. Martin Volk

Department of Computational Linguistics

Faculty of Arts and Social Sciences

Department of Informatics

Faculty of Business, Economics and Informatics

Submission date: 02.02.2022

Abstract

This thesis presents work on novel machine translation (MT) systems between spoken languages and signed languages, represented in a sign language writing notation system, i.e., SignWriting. It seeks to address the lack of support for signed languages in current MT systems and research. Our research is based on the SignBank dataset, which contains pairs of spoken language text and signed language content in the Formal SignWriting (FSW) format. Novel methods are introduced to parse, factorize, decode, and evaluate FSW. Preprocessed data is then used in three major sets of experiments/models, leveraging a factored Transformer neural machine translation architecture. A bilingual setup translating from American Sign Language to American English achieves over 30 BLEU score, while two multilingual ones translating both directions between spoken languages and signed languages achieve over 20 BLEU score. We find that common MT techniques used to improve spoken language translation have a similar effect on the performance of sign language translation. We thus support the claim of including signed languages in natural language processing (NLP) research.

Zusammenfassung

In dieser Arbeit werden neuartige maschinelle Übersetzungssysteme (MÜ) zwischen gesprochenen Sprachen und Gebärdensprachen vorgestellt. Die Gebärdensprache ist dabei in SignWriting dargestellt, einem Schriftsystem für Gebärdensprachen.

Die Arbeit zielt darauf ab, die fehlende Unterstützung für Gebärdensprachen in aktuellen MÜ-Systemen zu beheben und auch zu mehr Forschung in diese Richtung zu animieren. Unsere Forschung basiert auf dem SignBank-Datensatz, der Paare von gesprochenem Text und Gebärdensprachinhalten im Formal SignWriting-Format (FSW) enthält. Es werden neuartige Methoden zum parsen, faktorisieren, generieren und evaluieren von FSW vorgestellt. Die vorverarbeiteten Daten werden dann in drei Haupt-Experimenten verwendet, wobei eine faktorisierte Transformer-Architektur für maschinelle Übersetzung zum Einsatz kommt.

Ein bilinguales System, das von der amerikanischen Gebärdensprache ins amerikanische Englisch übersetzt, erreicht über 30 BLEU, während zwei mehrsprachige Systeme, die in beide Richtungen zwischen gesprochenen Sprachen und Gebärdensprachen übersetzen, über 20 BLEU erreichen. Wir stellen fest, dass gängige MÜ-Techniken, die zur Verbesserung der Übersetzung gesprochener Sprachen eingesetzt werden, eine ähnliche Wirkung auf die Qualität der Gebärdensprachübersetzung haben. Wir unterstützen daher die Forderung, Gebärdensprachen in die Forschung zur Verarbeitung natürlicher Sprache (NLP) einzubeziehen.

Acknowledgement

First, I would like to thank many people from the Department of Computational Linguistics (CL). It is you that make this research and thesis possible.

Thank you, Simon Clematide, Phillip Ströbel, and Tilia Ellendorff, for the course Machine Learning for Natural Language Processing, which opened the world of ML/NLP for me. Thank you, Prof. Rico Sennrich and Chantal Amrhein, for the course Advanced Techniques of Machine Translation, which led me to the road of MT. Thank you, Sarah Ebling, for the guest lecture on sign language processing (SLP), which got me interested in signed languages.

Thank you, Michi Amsler, for supporting me on my Master's project, and encouraging me to do computational linguistic research. Thank you, Prof. Rico Sennrich, for recommending me for this Master's thesis project.

I want to thank my supervisors and advisors. Thank you, Prof. Martin Volk, for giving me, an Informatics student, the opportunity to do my thesis in the Department of CL. Thank you, Mathias Müller, for taking care of all the things around, giving me practical advice on coding, model training and scientific paper writing, and meticulous note-taking during our every meeting. Thank you, Amit Moryossef, for motivating me to do this research project, providing me with the initial proposal, prototype, and data, coming up with numerous great ideas during the research process, and offering me precious suggestions on the thesis.

In addition, I thank all colleagues from both UZH and ETH and the ML/NLP/MT community all over the world. I am truly working on top of your shoulders. Special thanks to those colleagues who helped review and proofread my work: Chao Feng, Xiaozhe Yao, Raphael Merx, and Colin Leong.

Lastly, deep thanks to my wife for her constant companionship, and to my parents for their quiet support from the other side of the earth.

Thank you so much to all. Eternal glory is yours.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

ASL	American Sign Language
BLEU	Bilingual Evaluation Understudy
BPE	Byte Pair Encoding
CHRF	Character N-gram F-score
COLAB	(Google) Colaboratory
DGS	German Sign Language (Deutsche Gebärdensprache)
DSGS	Swiss-German Sign Language (Deutschschweizer Gebärdensprache)
FSW	Formal SignWriting
MAE	Mean Absolute Error
MT	Machine Translation
NER	Named Entity Recognition
NLP	Natural Language Processing
NMT	Neural Machine Translation
RNN	Recurrent Neural Network
SLP	Sign Language Processing
SLT	Sign Language Translation
SMT	Statistical Machine Translation
SWU	SignWriting in Unicode

1 Introduction

1.1 Motivation

Machine translation technology is (re-)building the Tower of Babel[1] that has been dreamed of by the human race since a thousand years ago. The rise of mature neural machine translation systems, such as Google Translate[2] and DeepL Translator[3], has changed the way people communicate with each other both online and offline.

However, most of the current machine translation systems support only spoken language input and output (text and/or speech), which excludes around 200 different sign(ed) languages used by up to 70 million deaf people[4] from modern language technology.

As argued by Yin et al. (2021), we should include sign language processing (SLP) as an integral part of natural language processing (NLP), and likewise, include sign language translation (SLT) as an integral part of machine translation (MT), for sign languages are also natural languages.

From a technical point of view, SLP/SLT brings novel challenges to NLP/MT due to sign languages' visual-gestural modality and special linguistic features (such as the use of space, simultaneity, and referencing), which requires both computer vision (CV) and NLP technologies. On the CV side, significant research aims to process sign language videos into poses[5] (using pose estimation) and gloss annotations[6], which is reviewed and summarized in Yin et al. (2021) and Moryossef and Goldberg (2021).

We believe that SLT, if formulated as an end-to-end task, involves translation between video and text (or video and video). This is challenging for several reasons.

[1] Tower of Babel: `https://en.wikipedia.org/wiki/Tower_of_Babel`

[2] Google Translate: `https://translate.google.com/`

[3] DeepL Translator: `https://www.deepl.com/translator`

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

[5] Poses: `https://en.wikipedia.org/wiki/Pose_(computer_vision)`

[6] Gloss: `https://en.wikipedia.org/wiki/Gloss_(annotation)#In_linguistics`

One of them is that video input is high-dimensional, and not all parts of a video are relevant in a linguistic sense. Another is that there is no standardized or widely used written form (Moryossef and Goldberg, 2021). This has hindered the inclusion of signed languages in NLP research.

However, some writing notation systems do exist, for instance SignWriting[7] (Sutton, 1990) and HamNoSys[8] (Prillwitz and Zienert, 1990). Figure 1 illustrates each sign language representation and their relationship - pose, gloss and writing notation.

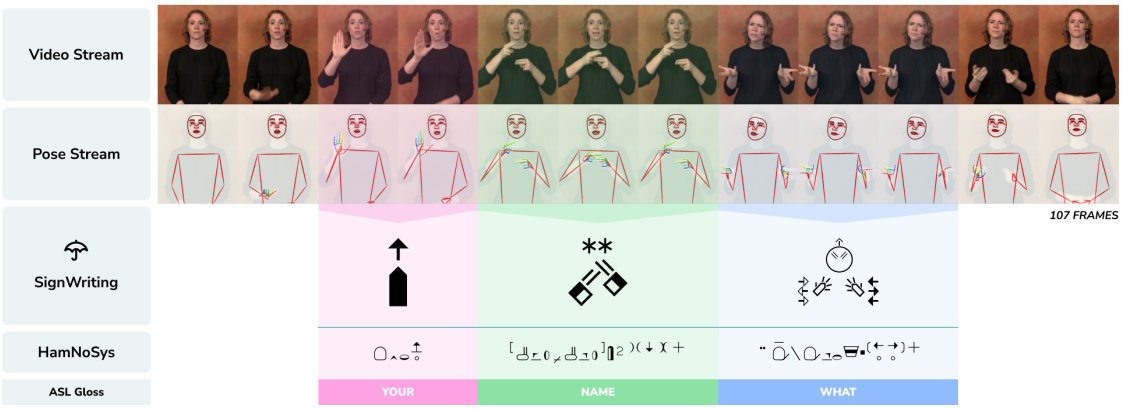


Figure 1: Representations of an American Sign Language phrase with video frames, pose estimations, SignWriting, HamNoSys and glosses. English translation: "What is your name?". Figure from Yin et al. (2021).

From Moryossef and Goldberg (2021), there is currently no research on the translation task between a sign language writing notation system and any other modality. We propose to formulate SLT tasks as follows:

Video input → Convert to a writing notation → Translate to spoken language text,

and the reverse direction:

Spoken language text → Translate to a writing notation → Convert to video output.

A conceivable scenario is translating between a sign language writing notation system and spoken languages. We choose SignWriting as the writing notation system for our research because:

- It is universal (multilingual), comparatively easy to understand, extensively documented, and computer-supported.
- Despite looking pictographic, it is a well-defined writing system. Every sign is

[7]SignWriting: `https://signwriting.org/`

[8]HamNoSys: `https://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/hamnosys-97.html`

written as a sequence of symbols/graphemes, and their location, which means it fits well into the current MT pipelines.

1.2 Research Questions

In this thesis, we explore the possibility and quality of creating bidirectional and multilingual machine translation systems to translate between spoken languages and signed languages in written form. We research how SignWriting, a writing notation system of signed languages, can be represented in a machine-readable format given its pictographic nature. Finally, we investigate to what extent training and evaluation methodologies used in spoken language translation research can be applied to sign language translation.

1.3 Thesis Structure

In Chapter 1, we have introduced the motivation (§1.1) behind research on translation between spoken languages and signed languages, represented in SignWriting, a sign language writing notation system. We have also listed the research questions (§1.2) that we are concerned about.

Chapter 2 introduces the background necessary to understand the research presented in the thesis, including Machine Translation (§2.1), Sign Language Processing (§2.2) and the SignWriting notation system (§2.3). The intended target audiences are readers who are familiar with NLP, but not necessarily with MT and/or signed languages.

The following three chapters present the main body of research on MT systems between spoken and signed languages, each with a different focus. Chapter 3 focuses on the data used for this research and the methods used to process it. Chapter 4 sets up three major groups of experiments/models - the first one is bilingual from American Sign Language to American English, while the other two are multilingual in both directions between spoken and signed languages. Chapter 5 presents and discusses the results of the experiments and lists some possible avenues for future research (§5.4).

Finally, Chapter 6 summarizes the main findings from this research.

2 Background

2.1 Machine Translation

MT is a subfield of computational linguistics that investigates the use of software to translate utterances from one language to another. Utterances can include text, speech as well as videos of signed languages.

In this research, we focus on translation between text (of spoken languages) and text (of signed languages). Throughout history, different paradigms and techniques have shaped MT (Hutchins, 1995, 2006). They are summarized in this section.

2.1.1 From Statistical to Neural Models

Statistical machine translation (SMT) (Koehn, 2009) is a machine translation paradigm where translation output is generated based on statistical models. The parameters of statistical models are derived from the analysis of bilingual text corpora.

SMT systems are essentially counting-based. The features for counting are usually shallow features found in parallel corpora, e.g., word frequency. One of the main challenges of SMT is how to estimate the probability of unseen words/n-grams [1].

Neural machine translation (NMT) (Koehn, 2017) is a machine translation approach that uses an artificial neural network to predict the likelihood of a sequence of words, which usually models entire sentences in a single integrated model.

NMT systems, as opposed to SMT systems, do not rely directly on counting features. Instead, NMT systems rely on complex architectures that are often black-box to humans.

At the time of writing, NMT systems are the predominant kind of MT systems. A typical NMT system includes the following components:

[1] N-gram: `https://en.wikipedia.org/wiki/N-gram`

1. Take in words/tokens as vectors by one-hot[2] encoding and/or word embedding (Mikolov et al., 2013).
2. Send input vectors to a recurrent neural network (RNN) language model (Mikolov et al., 2010), or an attentional encoder-decoder architecture (Bahdanau et al., 2015).
3. Output the probability distribution on target words/tokens through a softmax[3] activation function, which produces a cross-entropy[4] loss.
4. Gradient-based optimization is applied since mathematical functions in the network are generally differentiable and can be backpropagated[5].
5. During inference, beam search (Freitag and Al-Onaizan, 2017) is used to efficiently and approximately find the best hypothesis.

2.1.2 Preprocessing Techniques: Tokenization and Segmentation

Tokenization and segmentation are very common preprocessing techniques for MT systems. Tokenization is essentially splitting sentences into smaller units, such as individual words. Each of these smaller units is also called a token.

After tokenization, another common preprocessing step is subword-level segmentation, which allows open-vocabulary NMT translation with a fixed-size vocabulary. Byte pair encoding (BPE) is a commonly used algorithm for subword-level segmentation, as proposed by Sennrich et al. (2016a).

After translation, necessary postprocessing steps are applied to undo tokenization and segmentation.

2.1.3 Transformer Architecture

Published by Vaswani et al. (2017), Transformer (see Figure 2) is a new state-of-art architecture for NMT. Transformer models are superior (to recurrent models) in translation quality while being more parallelizable and requiring significantly less time to train.

[2]One-hot: `https://en.wikipedia.org/wiki/One-hot#Natural_language_processing`
[3]Softmax function: `https://en.wikipedia.org/wiki/Softmax_function`
[4]Cross-entropy: `https://en.wikipedia.org/wiki/Cross_entropy`
[5]Backpropagation: `https://en.wikipedia.org/wiki/Backpropagation`

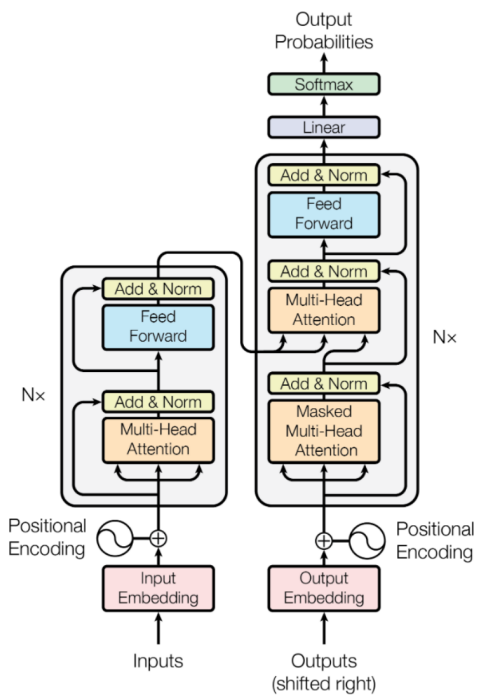


Figure 2: Transformer architecture. Diagram shows the encoder part on the left and decoder part on the right. Figure from Vaswani et al. (2017).

In the Transformer architecture, not only are the encoder and decoder connected through an attention mechanism, RNN layers in traditional encoder-decoder architecture are entirely replaced by multiple self-attentional layers[6]. These self-attentional layers are called Transformer blocks, connected by residual connections (He et al., 2016).

2.1.4 Multilingual Translation Systems

Under the assumption that a language-independent abstract layer called "Interlingua" (Rikters, 2019) can be shared across languages, multilingual translation systems have been developed over recent years.

The first fully shared model was proposed by Johnson et al. (2017). It does not introduce a new system architecture, only relying on adding an artificial token/tag at the beginning of the input sentence to specify the required target language.

Massively multilingual systems (including up to 100 languages) are researched by

[6]"Attention is all you need."

Aharoni et al. (2019) and Zhang et al. (2020), where zero-shot translation is also explored.

2.2 Sign Language Processing

SLP (Bragg et al., 2019; Yin et al., 2021; Moryossef and Goldberg, 2021) is an emerging subfield of both NLP and CV. Related research focuses on automatic processing and analysis of sign language content.

2.2.1 Signed Languages

Signed languages (also known as sign languages) are the primary means of communication for many deaf and hard-of-hearing people. They use the visual-manual modality to convey meaning, including manual articulations (hands, arms) and non-manual elements (shoulders, head, face). Signed languages are full-fledged natural languages with their own grammar and lexicon (Sandler and Lillo-Martin, 2006).

Signed languages are not universal and are usually not mutually intelligible with each other, although there are similarities among different signed languages. Signed languages do not rely on spoken languages, i.e., American Sign Language (ASL) is not a visual form of English, but its own unique language.

Nevertheless, there are sometimes mappings between spoken languages and signed languages. For example, in Switzerland, there are three signed languages - Swiss German Sign Language (DSGS), French Sign Language of Switzerland, and Italian Sign Language of Switzerland, as Swiss German, French and Italian are three major spoken languages in Switzerland. Furthermore, DSGS has six major dialects (developed in Zürich, Bern, Basel, Lucerne, St. Gallen, and Liechtenstein), which exemplifies the sparsity of signed languages.

2.2.2 Sign Language Representations

Signed languages can be represented in different ways (Yin et al., 2021; Moryossef and Goldberg, 2021), as illustrated by Figure 1. The most straightforward way is through video recording, to capture the visual-gestural modality. However, video files are large, making them expensive to store and transmit. Videos are also too high-dimensional, meaning not all parts of a video are relevant in a linguistic sense.

A lower-dimensional and more lightweight representation is the human pose. CV technique like pose estimation can be applied to derive poses from videos (Pishchulin et al., 2012; Chen et al., 2017; Cao et al., 2021; Güler et al., 2018), which, if done accurately, can encode all the relevant information for SLP.

Another representation that can be transcribed from videos is glossing. Every gloss is a unique text-form identifier that corresponds to a specific sign. A disadvantage is that glosses are bound to the semantics of signs, so they are language-specific. Unlike videos and poses, a new glossary is needed for every signed language.

A more universal solution is a writing notation system that writes down signed languages as pictographic symbols. This can be done either linearly (HamNoSys, Prillwitz and Zienert (1990)) or in two dimensions (SignWriting, Sutton (1990)). However, neither of them has been adopted widely enough to become the standard written form of any signed language. Nevertheless, an advantage of a writing notation system over poses is that it matches more closely the current spoken language processing pipelines.

2.2.3 Sign Language Processing Tasks

Among the most widely researched SLP tasks (Moryossef and Goldberg, 2021), we find: 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, many tasks including sign language recognition (Adaloglou et al., 2021), translation, and production consist in transforming one representation of sign language to another, as shown in Figure 3.

The focus of this research is on translation between a sign language writing notation system, i.e., SignWriting, and spoken language text, where we can try to exploit the current MT techniques used to translate spoken languages, bypassing the CV processing of videos.

2.2.4 Fingerspelling

Fingerspelling (Battison, 1978; Wilcox, 1992; Brentari and Padden, 2001) is an interesting linguistic phenomenon where a signed language meets a related spoken language written form. Sign language users borrow a word of the spoken language by spelling it letter-by-letter by finger. Fingerspelling is usually used on named

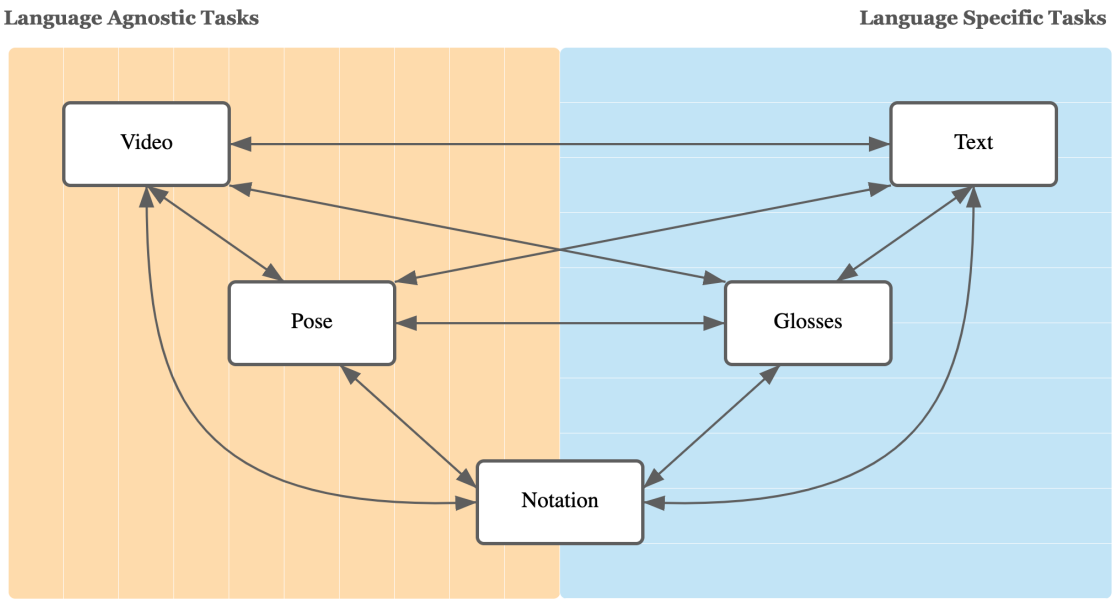


Figure 3: Sign language processing tasks. Every edge to the left, on the orange background (left side, background with vertical lines), represents a task in computer vision. These tasks are inherently language-agnostic, thus generalizing between signed languages. Every edge to the right, on the blue background (right side, background with horizontal lines), represents a task in natural language processing. These tasks are sign language-specific, requiring a specific sign language lexicon or spoken language tokens. Every edge on both backgrounds represents a task requiring a combination of computer vision and natural language processing. Figure from Moryossef and Goldberg (2021).

entities, such as a person, location, organization, etc. In SignWriting, fingerspelling is marked with a closed set of hand shapes.

2.2.5 Existing Datasets

At the time of writing, there is no publicly agreed way of sharing and loading SLP datasets. Thanks to Moryossef and Goldberg (2021), a library called Sign Language Datasets (Moryossef, 2021) can be used to load some existing sign language datasets, such as:

- the RWTH-PHOENIX-Weather 2014 T dataset (Cihan Camgöz et al., 2018) of German Sign Language,
- the Public DGS Corpus (Hanke et al., 2020) of German Sign Language,
- the multilingual SignBank dataset, which is introduced in subsection 2.3.3 and used throughout this research.

2.3 SignWriting

SignWriting (Sutton, 1990) is a sign language writing notation system. It was developed by Valerie Sutton[7], and is currently managed by Steve Slevinski[8].

SignWriting is very featural and visually iconic, both in:

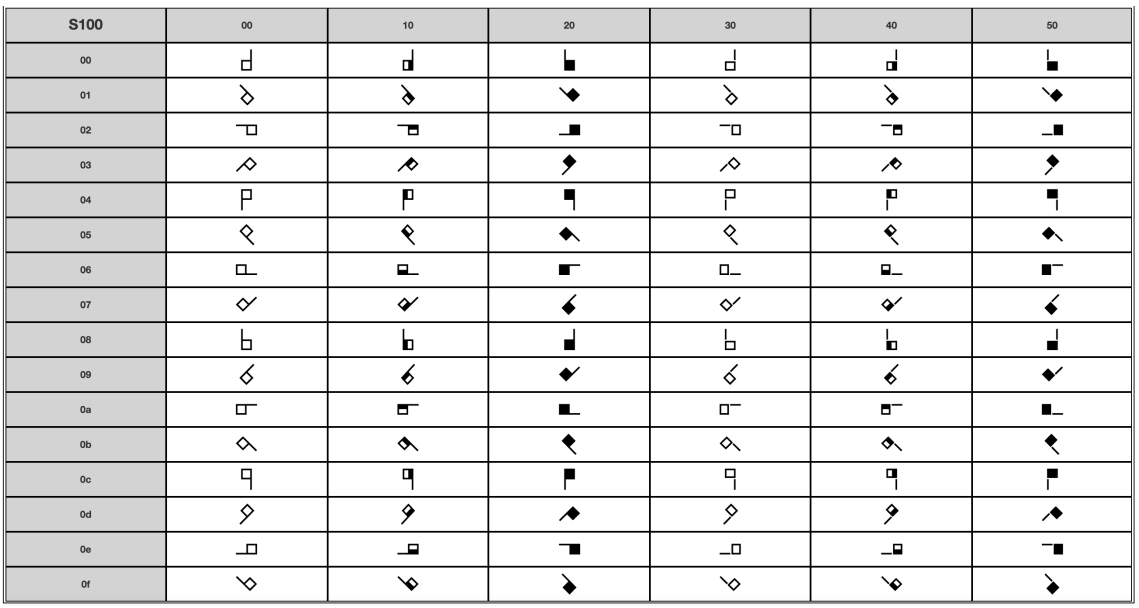
- the shapes of the symbols/graphemes, which are abstract pictures of hand shapes (see Figure 4), orientation (see Figure 5), body locations, facial expressions, contacts, and movement,
- the symbols' two-dimensional spatial arrangement within an invisible "sign box" (see Figure 6).

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 shifts of the body.

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

[8]Steve Slevinski: `https://steveslevinski.me/`

Figure 4: Hand shapes (some of) and their equivalents in SignWriting. Figure from Wikipedia.



S100	00	10	20	30	40	50
00						
01						
02						
03						
04						
05						
06						
07						
08						
09						
0a						
0b						
0c						
0d						
0e						
0f						

Figure 5: 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 also be seen as a factorization of the symbol.

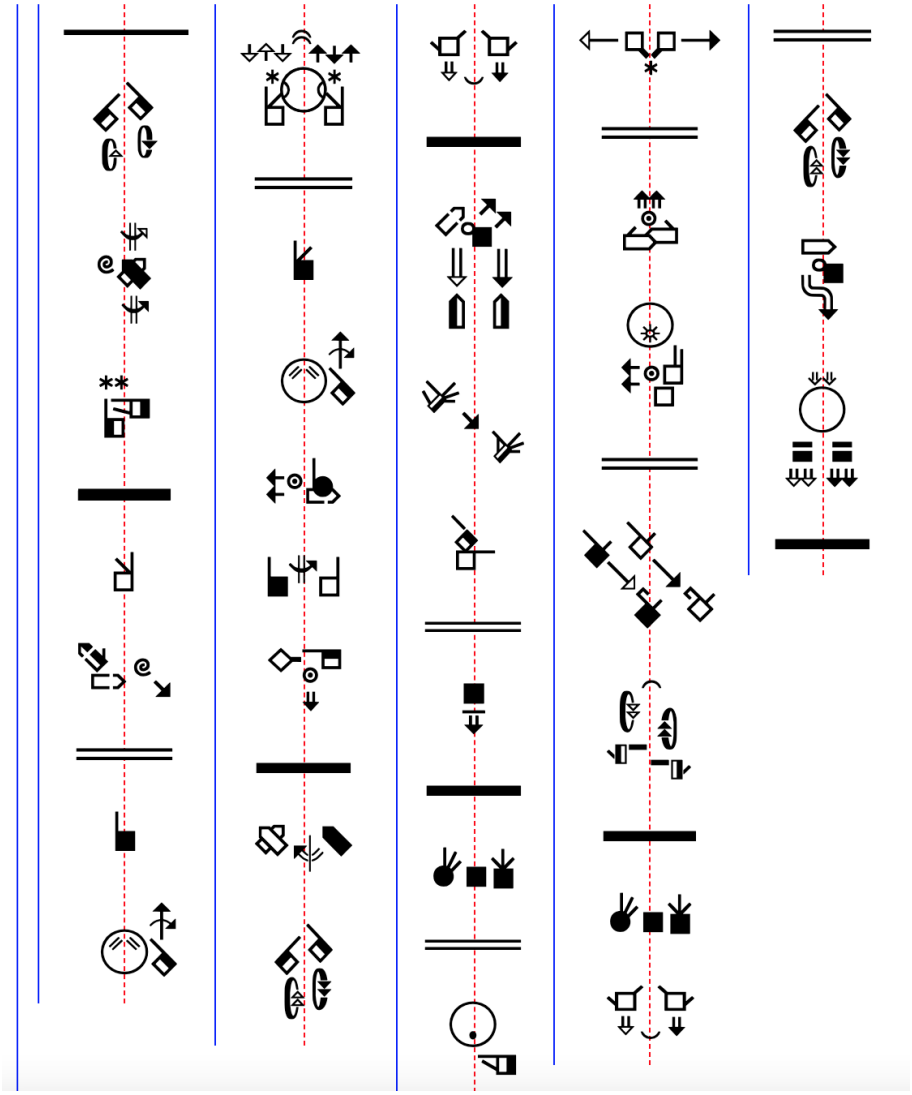


Figure 6: An example of SignWriting written in columns, the ASL translation of an introduction of Formal SignWriting in ASCII (FSW). The relative positions of the symbols within the box iconically represent the locations of the hands and other parts of the body involved in the sign being represented.

2.3.1 Formal SignWriting in ASCII (FSW)

In 2012, the Formal SignWriting in ASCII (FSW) specification (Slevinski, 2021) was released, which is a faithful encoding of SignWriting, documented in an Internet Draft submitted to the IETF[9].

The design of FSW is computerized, so it can be recognized and processed by programs (e.g., regular expressions). While signed languages are natural languages, FSW is a formal language, which means FSW is very handy in mathematics, computer science, and linguistics.

Although SignWriting is two-dimensional, FSW is written linearly like spoken languages. Each sign (box) is written as first a box mark, then a sequence of symbols, and their relative position (notated by positional numbers x and y), as illustrated by Figure 7.

2.3.2 SignWriting in Unicode (SWU)

In 2017, the SignWriting in Unicode (SWU) specification (Slevinski, 2021) was released, making SignWriting the first sign language writing system to be included in the Unicode Standard. The Unicode block for SWU is `U+1D800 - U+1DAAF`.

As illustrated in Figure 7, SWU is also written linearly. FSW and SWU are isomorphic and interchangeable, and both encode the complete information of SignWriting, which means we can use either of them for SLP and SLT.

2.3.3 SignPuddle and SignBank

SignPuddle is a community-driven dictionary where people add parallel SignWriting and spoken language text (usually dictionary pairs, sometimes sentence pairs). Each such dictionary is called a puddle. The community-driven aspect makes them multilingual but very unbalanced among languages, and a bit noisy. All the data is open-source.

SignBank is the largest repository of SignPuddles[10], with varying degrees of semantic separation, e.g., different signed languages, or different domains (literature vs the Bible).

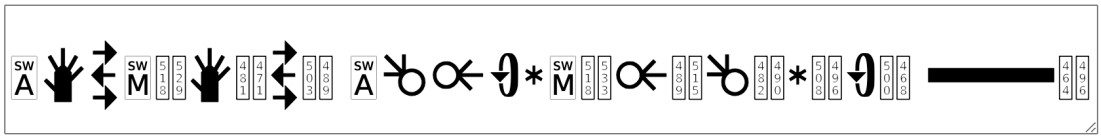
[9] Formal SignWriting: `https://www.ietf.org/id/draft-slevinski-formal-signwriting-08.html`

[10] SignBank and SignPuddle: `https://www.signbank.org/signpuddle/`

Formal SignWriting in ASCII

```
AS14c20S27106M518x529S14c20481x471S27106503x489
AS18701S1870aS2e734S20500M518x533S1870a489x515S18701482x490S20500508x496S2e734500x468
S38800464x496
```

SignWriting in Unicode



SignWriting Graphics

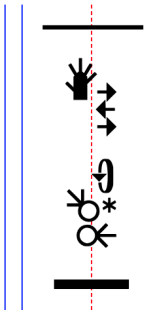


Figure 7: ASL "Hello world." in FSW, SWU and SignWriting graphics. In FSW/SWU, `A`/`SWA` and `M`/`SWM` are the box marks, `S14c20` and `S27106` (graphics in SWU) are the symbols, `518` and `529` are the positional numbers, `S38800` (horizontal bold line in SWU) is the special full stop symbol.

3 Data and Method

The data source we use for this research is the Sign Language Datasets (Moryossef, 2021). This library lets users load available sign language datasets using Tensorflow Datasets[1], including the above-mentioned SignBank dataset.

Each row in the SignBank dataset is represented in the format as shown in Listing 3.1.

```
{
 'assumed_spoken_language_code': 'de',
 'country_code': 'ch',
 'id': '1755',
 'puddle': '48',
 'sign_writing': '
   M550x535S32a00482x483S15d09455x499S15d01522x497S22114516x484
    S22114456x484S20f00524x522S20f00451x523',
 'terms': ['Kosmetiker', 'S4-05682']
}
```

Listing 3.1: An example of a data row in SignBank dataset

Where:

- `id` is the row identifier, and `puddle` denotes which SignPuddle the row belongs to.
- `assumed_spoken_language_code` denotes the assumed[2] spoken language, and `country_code` denotes from which country the signed language originates. `country_code` is not unique - `ch` (Switzerland) has multiple puddles with `it` (Italian), `fr` (French) and `de` (German) spoken languages. `assumed_spoken_language_code` and `country_code` together form a one-to-one mapping to one signed language, although we do not have an ideally explicit `sign_language_code` in this dataset.
- `sign_writing` is the sign language text in FSW format.

[1] Tensorflow Datasets: `https://www.tensorflow.org/datasets`

[2] The author of the library assumed the spoken languages when creating the dataset from the SignBank website.

- `terms` is an array of spoken language text, which could contain the title, the main body, and other additional information. We have to extract the meaningful part that is corresponding to the FSW text.

Our goal is to form a multilingual parallel text translation corpus between FSW and spoken language text, to train translation models.

3.1 Data Statistics

Before curating the data, we first collected some data statistics to gain an overall understanding of the SignBank dataset.

There are $\sim$ 220k parallel samples in 141 puddles in 76 language pairs (by spoken language + country code combination), yet the distribution is very unbalanced, as illustrated by Figure 8 and Figure 9.

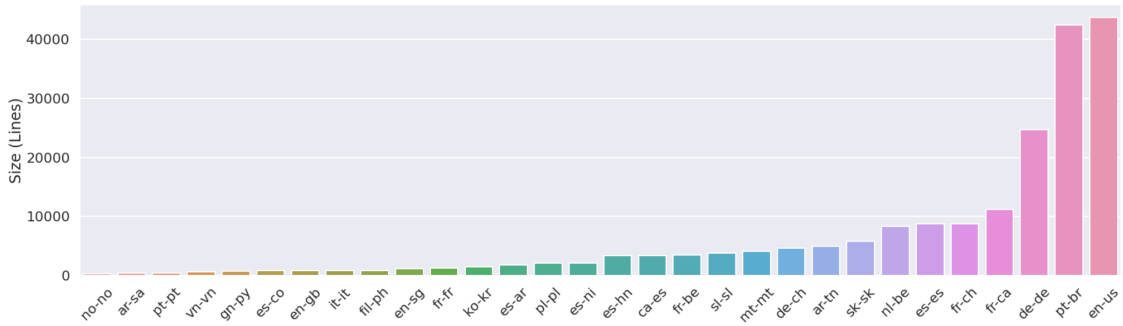


Figure 8: Data distribution (the first 30 language pairs). Figure plotted by Mathias Müller's Colab notebook.

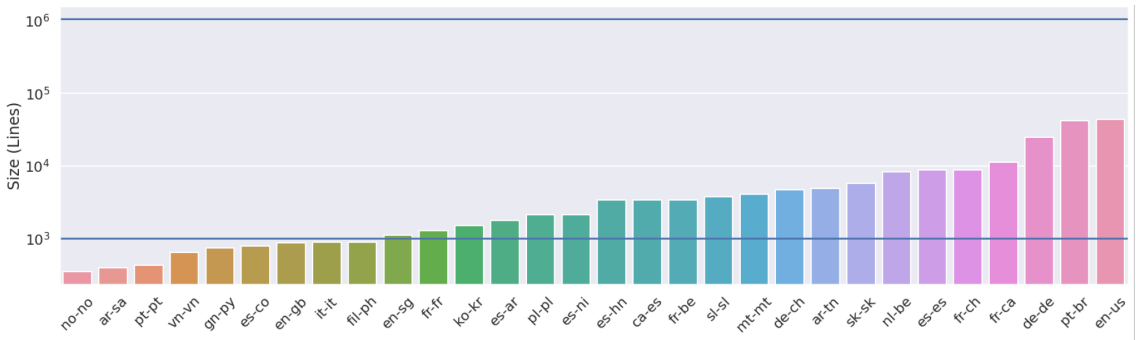


Figure 9: Data distribution (the first 30 language pairs) in log scale. Figure plotted by Mathias Müller's Colab notebook.

Relatively high-resource language pairs (over 10k samples) are listed in Table 1.

language pair	#samples	#puddles
en-us (American English & American Sign Language)	43698	7
pt-br (Brazilian Portuguese & Brazilian Sign Language)	42454	3
de-de (Standard German & German Sign Language)	24704	3
fr-ca (Canadian French & Quebec Sign Language)	11189	3

Table 1: Relatively high-resource language pair statistics

See Table 7 for an exhaustive list of all 21 (spoken) languages involved in this research.

Another important fact is that most of the puddles are dictionaries, which are not as good resources as sentence pairs for training translation systems, because by only learning from dictionaries, models cannot learn how to formulate sentences, instead, they just memorize word mappings.

Thus, we treat the four sentence-pair puddles (see Table 2) of the relatively high-resource language pairs as primary data and the other dictionary puddles as auxiliary data.

puddle id	puddle name	language pair	#samples
5	Literature US	en-us	700
151	ASL Bible Books NLT	en-us	11667
152	ASL Bible Books Shores Deaf Church	en-us	4321
114	Literatura Brasil	pt-br	1884

Table 2: Primary sentence-pair puddles

Even though these language pairs constitute the high-resource pairs of the SignBank dataset, they are low-resource compared to the datasets used in mature machine translation systems for spoken languages, where millions of parallel sentences are commonplace (Bojar et al., 2014, 2015).

Positional numbers are an important component of FSW, for they determine how symbols are assembled in the two-dimensional sign box (see Figure 7). It is also interesting to look at how those positional numbers are distributed in Figure 10.

We can see that those numbers are fairly normally distributed around the center 500, and there are not many outliers.

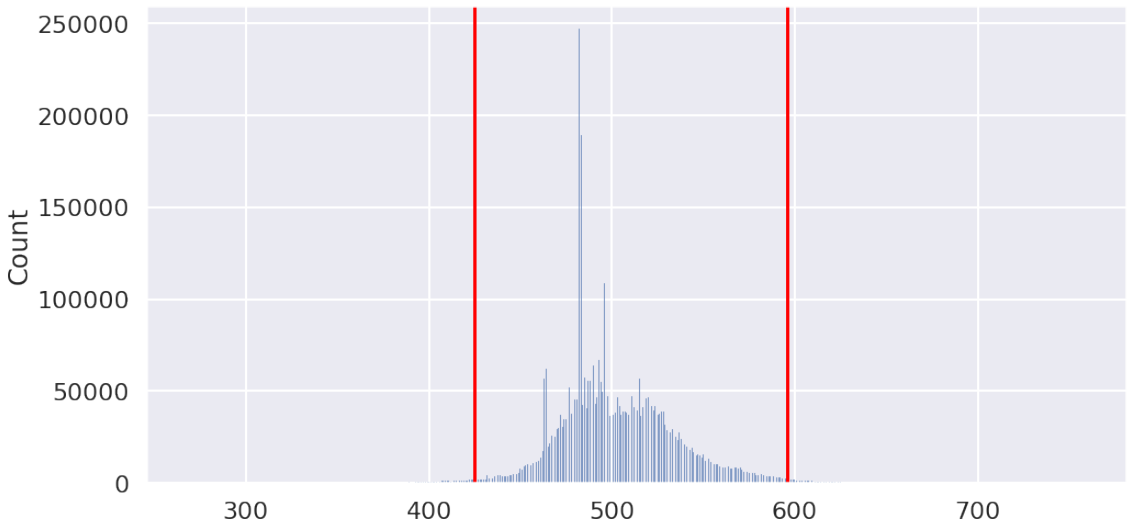


Figure 10: Positional numbers distribution. The red lines mean the first and 99th percentile. Figure plotted by Mathias Müller's Colab notebook.

3.2 Data Cleaning

Since the data source is diverse (141 different puddles) and mainly comes from community contributions, some data cleaning needs to be applied in the first place:

- Extract the main body of the spoken language text.
- Escape line-break characters.
- In the case of Bible-domain data (see subsection 2.3.3), remove Bible source information that is not relevant in FSW, e.g., "Nehemiah 3v11 NL".
- Remove empty samples.
- Remove samples that contain HTML tags, e.g., "<iframe>".
- Remove samples that contain extremely long sequences.

Note that those steps are rather general and mainly targeted at the most primary data (see Table 2).

3.3 BPE Segmentation on Spoken Languages

BPE subword-level segmentation (introduced in subsection 2.1.2) is widely used for preprocessing spoken language data in NLP tasks.

This technique is supposed to improve the performance in a low-resource setting like ours because it allows generalizing a small vocabulary (2000 in our case) to unseen words (Sennrich et al., 2016a).

In multilingual experiments (see subsection 4.2.2 and subsection 4.2.3), we applied shared vocabulary BPE segmentation to get more consistent segmentation across spoken languages and share possible common subword units. We should note that word sharing between spoken languages in our case is very common and frequent. For example, Canadian French (fr-ca) is not only similar to French French (fr-fr) but also similar to American English (en-us).

3.4 Parse FSW

Likewise, an appropriate segmentation/tokenization strategy is needed for the FSW data. To do this, we first parsed FSW into several pieces:

- box marks: `A`, `M`, `L`, `R`, `B`,
- symbols: `S1f010`, `S18720`, etc.,
- positional numbers x and y: `515`, `483`, etc.,
- punctuation marks (special symbols without box marks): `S38800`, etc.

We further factorized each symbol (take `S1f010` for example) into:

- symbol core: `S1f0` (see Figure 5 to understand how factorization works),
- column number (from 0 to 5): `1`,
- row number (from 0 to F): `0`.

For positional numbers, we further calculated two advanced features/factors that denote a symbol's relative position (based on the absolute numbers) within a sign:

- relative x: from 0 to the number of symbols in the sign - 1
- relative y: from 0 to the number of symbols in the sign - 1

3.5 Factored Machine Translation

After parsing FSW, we have many pieces of information, which, if put together into one sequence, would be very long. However, very long sequences are not efficient for training machine translation models - more memory, training time, and decoding time (beam search) are required.

From another perspective, the most essential information units are the symbols/graphemes. The positional numbers, on the other hand, could be supposed as auxiliary information. Nevertheless, the numbers are necessary as they determine how symbols are assembled, and the same symbols could be combined in a different spatial way to convey a different meaning.

We propose to use a factored machine translation architecture (Koehn and Hoang, 2007; Garcia-Martinez et al., 2016) to solve this task. In past work, factored machine translation architecture was used to integrate additional annotation at the word-level of spoken languages — may it be linguistic markups or automatically generated word classes, e.g., lemmas, Part of Speech tag, tense, person, gender, and number (see Figure 11).

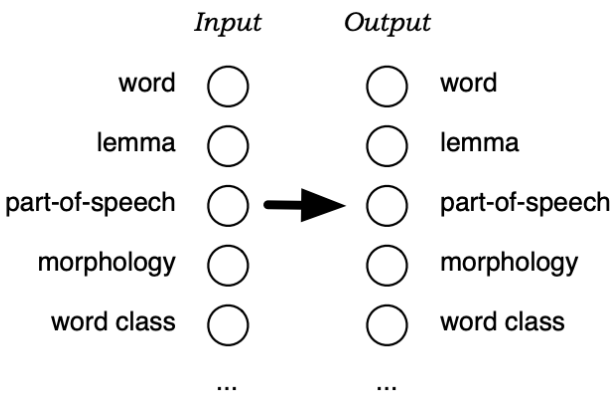


Figure 11: Factored representations of input and output words incorporate additional annotation. Figure taken from Koehn and Hoang (2007).

In our scenario, we treated the symbols (including the punctuation marks) and box marks (acting as sign boundaries) as the source/target tokens, and the following as source/target factors that are strictly aligned with source/target tokens:

- (absolute) positional numbers x, y
- relative positional numbers x, y
- symbol core

- symbol column number
- symbol row number

The FSW sentence from Listing 3.1 is factorized (and tokenized) in Listing 3.2.

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

Listing 3.2: An example of factorization

After factorization of FSW, `symbol` sequences and spoken language sequences can form source/target pairs for seq2seq (Sutskever et al., 2014) training, and the rest serve as (optional) factors.

At the encoding phase (when FSW is the source language), each input's embedding is the concatenation of each symbol's embedding and all the aligned factors' embedding.

At the decoding phase (when FSW is the target language), additional cross-entropy losses are calculated on the factors (advanced/factorized/relative ones not included since they are useless for prediction) and backpropagated, with a weight relative to the symbols' cross-entropy loss.

3.6 Multilingual Tags

As there are different languages on both signed and spoken language sides, to train multilingual models, we added language tags at the beginning of source sequences to indicate the desired target language.

As per Johnson et al. (2017), we applied this technique without changing the model architecture. Three types of tags were designed to encode all necessary information:

- spoken language code

- country code
- dictionary or sentence pair

The symbol sequence in Listing 3.2 is further prepended with the following tags:

```
<2de><4ch><dict> M S32a00 S15d09 S15d01 S22114 S22114 S20f00 S20f00
```

which means this FSW sequence is to be translated into Swiss-German (de-ch) as a dictionary word.

3.7 Data Enrichment: Automatic Translation from German

Inspired by the back-translation (Sennrich et al., 2016b) technique that is commonly used in MT, we experimented in enriching American English data by automatic translation.

We used DeepL API[3] to automatically translate German (de-de) to English (en-us), which raised the amount of American English (en-us) data from $\sim$ 40k to $\sim$ 60k, as indicated by Table 1.

3.8 Data Augmentation: Synthetic Fingerspelling Samples

Fingerspelling is often used in the SignBank dataset to express named entities, such as a person, location, organization, etc. Given the limited amount of fingerspelling samples in the dataset, another data augmentation technique we applied is to manually synthesize samples, to let models learn how to fingerspell Latin letters.

We collected a canonical mapping from letters to FSW symbols, as shown in Listing 3.3.

```
{
 'A': 'S1f720',
 'B': 'S14720',
 'C': 'S16d20',
 'D': 'S10120',
 ...
```

[3]DeepL API: https://www.deepl.com/en/docs-api/

```
}
```

Listing 3.3: Part of fingerspelling mapping

We then generated random length (from 1 to 10) sequence pairs by randomly choosing letters and symbols from the mapping. 10k randomly generated samples are thus added to the training data.

4 Experimental Setup

4.1 Data

The data used for our experiments is the above-mentioned SignBank dataset, preprocessed by the above-mentioned pipelines: cleaning, segmentation, parsing, factorization, tagging, enrichment, augmentation.

We split the data into training, validation, and test sets by shuffling the data and slicing 95%, 3%, and 2% respectively. Additional generated data (if any) is appended to the training set.

In the end, we had a split multilingual parallel corpus between FSW symbols and spoken language text, with multiple factors optional to use. Based on that, we wanted to train bidirectional translation models.

Note that not all the experiments used all the same data and method.

4.2 Models

4.2.1 40K SIGN to EN-US

The first group of models/experiments we did are bilingual, translating from FSW of American Sign Language to American English (en-us).

Data includes:

- $\sim$ 15k sentence pairs from 3 en-us puddles: Literature US, ASL Bible Books NLT, ASL Bible Books Shores Deaf Church,
- $\sim$ 25k dictionary pairs from 3 en-us puddles: Dictionary US, LLCN & SignTyp, ASL Bible Dictionary,
- $\sim$ 20k dictionary pairs from 1 de-de puddle: Wörterbuch DE (automatically translated to en-us),

which leads to:

- $\sim$ 6k source vocabulary size (number of non-factorized symbols),
- $\sim$ 2k target vocabulary size (determined by BPE).

We used a standard Transformer NMT model architecture (not pretrained), with a baseline configuration:

- 6 layers + 8 heads + 512 embedding size (0.0 dropout) + 512 hidden size (0.1 dropout) + 2048 feed forward size (0.1 dropout),
- initial learning rate 0.0001, decrease learning rate by a factor of 0.7 every 5 times validation score not improved,
- batch size 16 sentences, label smoothing 0.1, epochs 200,
- for testing, decoding with a checkpoint that has the best validation score, beam size 5, alpha for length penalty 1.

Then we trained and tuned several models with different data and hyperparameters combination until we reached the final configuration:

- 6 layers + 8 heads + 512 embedding size (16 for each factor) (0.5 dropout) + 512 hidden size (0.5 dropout) + 2048 feed forward size (0.5 dropout),
- batch size 32 sentences, label smoothing 0.2, epochs 300.

Note that the model architecture and most of the configuration stay unchanged, if not mentioned specifically in later experiments.

4.2.2 100K SIGN to SPOKEN

The second group of models/experiments extend the first from a bilingual setting to a multilingual setting, translating from FSW of multiple signed languages to spoken languages.

Data includes:

- $\sim$ 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),
- $\sim$ 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:

- $\sim$ 11k source vocabulary size (number of non-factorized symbols),
- $\sim$ 2k target vocabulary size (determined by BPE).

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

- batch size 4096 tokens.

We also raised the data amount to $\sim$ 170k pairs with more low-resource language pairs and puddles, including 21 language pairs that have over 1k samples (most of which are dictionaries) to train a massively multilingual model that sees as many languages from SignBank as possible.

4.2.3 100K SPOKEN to SIGN

The third group of models/experiments translate the reverse direction, from spoken languages to FSW of multiple signed languages, thus they are also multilingual.

Data is the same as the second: $\sim$ 17k sentence pairs + $\sim$ 83k dictionary pairs.

We tried to:

- predict everything (including positional numbers) as target tokens all in one very long target sequence, inspired by Chen et al. (2022),
- predict symbols only (as a comparative experiment),
- predict symbols with positional numbers as target factors (opposite to the models from FSW to spoken languages).

The model configuration remains the same.

4.3 Evaluation

4.3.1 BLEU, chrF and Perplexity

BLEU (Papineni et al., 2002) and chrF (Popović, 2015) scores are used as evaluation metrics on predicted sentences during testing. For chrF, specially chrF2 is used on

spoken languages, and chrF2++ is used on FSW symbols to evaluate on both word and character levels.

BLEU score is also used as the validation metric when translating from FSW to spoken languages. Perplexity (Chen and Goodman, 1996; Sennrich, 2012) is used as the validation metric instead when translating from spoken languages to FSW because BLEU's effectiveness was yet unknown on FSW.

4.3.2 Top-n Accuracy on Dictionary

Given that there are many dictionary pairs in the dataset, top-n (from 1 to 5) accuracy is calculated on dictionary translation, to reflect how good models learn to translate dictionary pairs.

4.3.3 Mean Absolute Error on Positional Numbers

Mean absolute error (MAE) is calculated to measure the distance between predicted positional numbers of FSW and golden samples:

$$\frac{1}{D}\sum_{i=1}^{D}|x_i - y_i|$$

4.3.4 Custom Fingerspelling Evaluation

We performed a custom evaluation to understand how well the models learn to fingerspell.

We ran a named entity tagger (StanfordNERTagger, Finkel et al. (2005)) to find all named entities in golden spoken language samples, and counted the accuracy of how many of them appear also in the predicted samples.

4.3.5 Side-by-side SignWriting Evaluation

Apart from all the quantitative evaluation metrics on spoken language and FSW text, it is essential to see how the graphical form of predicted SignWriting looks. After all, FSW is the written notation of sign language graphics.

We reconstructed FSW and its graphics to do a side-by-side evaluation to compare golden SignWriting samples and predicted SignWriting samples.

4.4 Tools

4.4.1 SentencePiece

SentencePiece (Kudo and Richardson, 2018) is used to do BPE segmentation on spoken languages. SentencePiece operates on raw sentences, so no other tokenization step is needed.

BPE vocabulary is shared across different spoken languages in multilingual settings. Vocabulary size is fixed to 2000.

4.4.2 Joey NMT

Joey NMT (Kreutzer et al., 2019) is the NMT framework used to do the first two groups of experiments/models from FSW to spoken languages.

As there was no factored machine translation support in Joey NMT at the time, we implemented the source factor function in Joey NMT, which allows multiple word-level source factors (to be either concatenated or summed to the input word's embeddings).

4.4.3 Sockeye

Sockeye (Hieber et al., 2020) is the NMT framework used to do the third group of experiments/models from spoken languages to FSW.

We chose Sockeye instead of Joey NMT for the convenience of ready-to-use target factor support (note that Sockeye also supports the source factor). Positional numbers of FSW can be easily added as target factors, which contribute additional cross-entropy losses to the training process.

4.4.4 SacreBLEU

SacreBLEU (Post, 2018) is used to calculate BLEU and chrF (chrF2, chrF2++) scores.

5 Results and Discussion

5.1 40K SIGN to EN-US

As being the first group of models (see subsection 4.2.1), several experiments were done to:

1. prove that translation from FSW of American Sign Language to American English (en-us) works,
2. find the best data and hyperparameter combination.

Table 3 shows the results of evaluation on test set.

expt. id	model	BLEU	chrF2
E1	baseline	22.5	
E2	full (1 + dicts)	25.2	
E3	bpe (2 + BPE)	27.0	46.2
E4	bpe_factor (3 + x,y as factors)	27.5	46.5
E5	bpe_factor_sign+ (4 + symbol core as source + row,col as factors) ☹	23.1	41.2
E6	bpe_factor_rel (4 + relative x,y as factors)	28.1	47.5
E7	bpe_factor_rel_tune (6 + aggressive dropout + tied softmax)	31.4	52.0
E8	bpe_factor_sign+ (7 + symbol core,row,col as factors) ☺	**32.0**	**52.7**
E9	bpe_factor_sign+ (8 + remove lowercasing) ☺	**30.8**	**51.2**
E10	bpe_factor_sign+ (9 + smaller BPE vocab 2000 to 1000) ☹	29.5	50.8
E11	bpe_factor_sign+_de (9 + de dict) ☹	29.9	50.6

Table 3: Results of 40k sign to en-us. Evaluated in BLEU (higher is better) and chrF2 (higher is better), and the best scores are set in bold and marked by ☺. Experiments that do not improve the metrics are marked by ☹. In the parentheses is the relationship between experiments.

5.1.1 Effect of Adding Dictionaries

First, as shown by the comparison between *E1* and *E2*, the effect of adding $\sim$ 25k dictionary pairs in addition to $\sim$ 15k sentence pairs is obvious. This improves the overall BLEU score by 2.7.

Based on this observation, we decided to always include available dictionary pairs in later experiments.

5.1.2 Effect of BPE

The effect of BPE segmentation on the spoken language's (American English) side is also obvious. Comparing *E2* and *E3*, there is a 1.8 BLEU score improvement.

Likewise, we always applied BPE segmentation on the spoken language's side in later experiments.

5.1.3 Utilize Positional Numbers

We applied novel methods to parse and factorize FSW as discussed in subsection 3.4 and subsection 3.5, yet it was unknown how to best utilize the positional numbers from a model training perspective.

In *E4*, we tried to add x and y positional numbers as source factors, which improves BLEU score by 0.5 and chrF2 score by 0.3 as compared to *E3*.

In *E5*, we did a further experiment on replacing symbols with symbol cores as the source, and made the row and column numbers together with x and y as source factors, under the assumption that symbol cores are the most essential information units, and the rows and columns, as well as x and y, are only possible variations applied to the cores.

However, the results show that this change worsens BLEU score by 4.4 and chrF2 score by 5.3, compared to *E4*. We speculated that the data is not high-resource enough for the model to learn all variations by columns and rows, thus making a `S10001` to be an `S100` loses certain information.

In *E6*, we added relative x and y positional numbers as source factors in addition to absolute x and y in *E4*, which improves the BLEU score by 0.6 and chrF2 by 1.0. This fact confirms our assumption that symbols' relative position in a sign is important, and explicitly encoding such information helps the translation model.

In *E8*, we added back symbol cores, columns, and rows all as source factors, while keeping the symbols as the source. It is redundant in terms of the information because symbol cores, columns, and rows are just factorizations of symbols. Nevertheless, as a result, *E8* further improves BLEU score by 0.6 and chrF2 score 0.7 by as compared to *E7*.

To conclude, the best strategy to utilize the positional numbers is explicitly adding all additional information as source factors - x, y, relative x, relative y, symbol core, column number, row number, while keeping symbols as main source tokens.

A further note is that the effect of utilizing positional numbers is not as large as we had expected. The symbols alone already encode very much information and work well as the source language.

5.1.4 Effect of Low-resource Tricks

Since it is a rather low-resource setting, we applied aggressive dropout and tied softmax (only tied between the embedding of the input and output of the decoder) as low-resource tricks (Sennrich and Zhang, 2019). See detailed configuration in subsection 4.2.1

As shown by the comparison between *E7* and *E6*, low-resource tricks borrowed from spoken language translation tasks prove to be effective on sign language translation as well - improve BLEU score by 3.3 and chrF2 score by 4.5.

5.1.5 Effect of Smaller BPE Vocabulary

We tuned the BPE vocabulary size in *E10*. Smaller BPE vocabulary (of size 1000) worsens BLEU score by 1.3 and chrF2 score by 0.4, as compared to *E9*. So we kept the BPE vocabulary size to 2000.

5.1.6 Effect of Adding Automatic Translated Samples

Finally, adding $\sim$ 20k automatic translated samples (from German to English, see section 3.7) to training set in *E11* proves not to be a good idea - worsens BLEU score by 0.9 and chrF2 score by 0.6, as compared to *E9*.

Possible reason: German Sign Language (DGS) is a totally different signed language from American Sign Language (ASL) in terms of how signs are signed and written

in SignWriting (see Figure 12), so mixing them does not help.

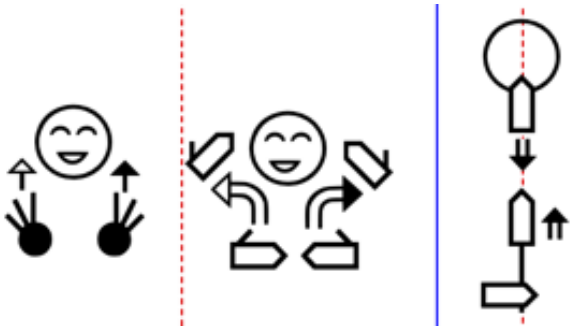


Figure 12: Sign language: DGS vs. ASL. Separated by the blue (solid) vertical line, on the left is "guten Morgen" signed in DGS, on the right is "good morning" signed in ASL. DGS tends to use the horizontal direction more than ASL.

5.2 100K SIGN to SPOKEN

In the second group of models/experiments (see subsection 4.2.2), we explored translation in a multilingual setting. The number of experiments is not as many, since we had almost fixed the data and hyperparameter combination during the first models/experiments. Conversely, the number of evaluation metrics rises because top-n accuracy is introduced to evaluate dictionary pairs, and specific evaluation on each language is applied.

Table 4 shows the results of evaluation on the test set (the orientation of the table changes to adjust to page width limit).

5.2.1 Multilingual Performance

Generally speaking, the more resources a language has in the multilingual model, the better its performance is (Zhou et al., 2021). The most frequent two target languages American English (en-us, 40k) and Brazilian Portuguese (pt-br, 40k) have shown good evaluation metrics, yet the rather poor performance on some low-resource languages is reasonable because most of them only have a limited amount of dictionary pairs.

There is evidence to show that a relatively high-resource language can help a related low-resource language if we look at the performance of Singaporean English (en-sg, 1k). Its accuracy is nearly as good as German German (de-de, 20k), which is very

language	metrics	4 language pairs 100k samples	21 language pairs 170k samples
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
es-hn (3k)	top-1		**0.14**
	top-5		**0.20**
es-ni (2k)	top-1		0.08
	top-5		0.13
es-es (8k)	top-1		0.02
	top-5		0.03

Table 4: Results of 100k sign to spoken (partial results on most frequent languages). Evaluated in BLEU (higher is better), chrF2 (higher is better) and top-n accuracy (higher is better), and good scores are set in bold. Languages without sentence pairs are only evaluated by top-n accuracy. For the first model many cells are left empty for languages that do not participate in the first model. In the parentheses are the rough numbers of samples per language.

likely improved by American English (en-us, 40k). Surprisingly, French French (fr-fr, 1k) and Honduran Spanish (es-hn, 3k) have also shown good performance under very low-resource conditions.

We should also note that the difference (theme, format, data cleanness, etc.) in puddles could result in evaluation performance differences between different languages. Moreover, a more fine-grained preprocessing step is needed for each puddle and each language to improve the performance of the long-tailed low-resource languages.

5.2.2 Curse of Multilinguality

Compared to the comparable best bilingual en-us model (*E9* from Table 3), en-us in the first multilingual model's BLEU score is 1.3 behind and chrF2 score is 1.4 behind, which is an acceptable drop.

However, when extending from 4 language pairs to 21 language pairs, a more severe drop appears - en-us drops 4.5 in BLEU score and 2.8 in chrF2, pt-br drops 17.4 in BLEU score and 26.8 in chrF2.

In a sign language translation setting, we have a similar observation as discussed in Aharoni et al. (2019) - average performance decreases with the number of languages. They suggest deeper and wider models could alleviate the problem.

5.2.3 A Word on Top-n Accuracy

It is tricky to evaluate the translation quality of dictionaries. For a given word, a model either has seen or has not seen the word during training, and it should memorize and predict well if it has seen the word, otherwise not, assuming the model has enough capacity.

Thus, how well a translation model performs on dictionary pairs depends much on how much testing data is overlapped within the training data. The paradox is - we can not guarantee a model to generalize well even if it gets good accuracy on the test set (what if the test set is overlapped with the training set?). Conversely, we cannot say a model is bad when it gets poor accuracy, maybe it is just because there is nothing similar in the test set as seen and memorized during training (although the model memorizes very well).

Anyway, if a model has seen all the words from a language, then it should perform well on whatever dictionary test set. However, it is not the case in our low-resource

setting.

5.2.4 Effect of Adding Synthetic Fingerspelling Data

Synthetic fingerspelling data is added to the training set for teaching models how to fingerspell, as discussed in subsection 3.8. The evaluation method of fingerspelling is discussed in subsection 4.3.4.

We did comparative experiments on the first multilingual (4 language pairs, 100k samples) model. The fingerspelling accuracy on en-us of the original data is 0.76, while the fingerspelling accuracy on en-us of the augmented data drops to 0.68. The synthetic fingerspelling data proves to be unhelpful.

A more sophisticated data augmentation strategy is needed, otherwise, other techniques should be applied (see subsection 5.4.1).

5.3 100K SPOKEN to SIGN

In the third group of models (see subsection 4.2.3), we explored translation in a reverse direction. The experiments remain in a multilingual setting (4 language pairs, 100k samples). The main challenge is to decode to FSW - both symbols and positional numbers.

As FSW is a universal representation across different signed languages and is more fine-grained (consisting of small graphemes/symbols) than spoken languages, we evaluated the overall BLEU and chrF2++ score on FSW symbols and evaluated additionally on positional numbers by the method discussed in subsection 4.3.3.

Table 5 shows the results of evaluation on test set.

5.3.1 Ways of Generating Positional Numbers

There are different ways of generating positional numbers. We first tried to treat them as normal target tokens in *E1*. However, the performance is poor - 6.6 BLEU score and 23.1 chrF2++ score (on symbol), which means the translation can hardly get the main idea of the source sentence. This also triggers the long sequence issue as discussed in subsection 3.5.

Instead, we treated positional numbers as target factors in the other experiments,

expt. id	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: Results of 100k spoken to sign. Evaluated in BLEU (on symbol, higher is better), chrF2++ (on symbol, higher is better), MAE (on positional numbers, lower is better), and the best scores are set in bold and marked by ☺. `w` denotes the weight between each factor's loss and the main target loss.

which all achieve over 20 BLEU score (on symbol).

At the decoding/testing phase, beam search is only applied based on symbol prediction. Target factors do not participate in beam search, i.e. each target factor prediction is the argmax of the corresponding output layer distribution.

5.3.2 Trade-off between Symbols and Positional Numbers

In *E2*, we translated only to the symbols as a baseline, then we tried out the translation with target factors with different weights.

We observed that the overall tendency is - the higher the weight is, the lower mean absolute error (MAE) on x and y is, the lower BLEU and chrF2++ scores on symbol are, and vice versa. *E6* (w=0.1) turns out to be the best trade-off point in between.

5.3.3 Possibly Flawed Positional Number Evaluation

Although the MAE looks reasonable and informative, we should note that it is possibly flawed, in the fact that the predicted symbol sequences can deviate from the golden symbol sequences. If this is the case, it is meaningless to do a token-by-token comparison on the positional numbers, even the sequence length can mismatch.

5.3.4 Multilingual Performance

Since the experiments remain in a multilingual setting, we performed a language-by-language evaluation to see the multilingual performance.

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

American Sign Language (en-us) performs the best as a relatively high-resource target language, so is German Sign Language (de-de). However, Brazilian Sign Language's (pt-br) performance drops unreasonably compared to that in Table 4. Unfortunately, the model somehow does not learn how to sign correctly in Brazilian Sign Language.

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: Results of 100k spoken to sign multilingual (w=0.1). Evaluated in BLEU (higher is better) , chrF2++ (higher is better), and the good scores are set in bold. In the parentheses are the rough numbers of samples per language.

5.3.5 Side-by-side SignWriting Evaluation

As promised in subsection 4.3.5, we performed a side-by-side evaluation of the graphics to gain some intuition on how the translation model signs.

Figure 13 shows the golden and predicted SignWriting graphics of American Sign Language corresponding to the American English sentence (from the Bible corpus):

"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."

As shown by the figure, at the beginning of the sentence, the model signs the same way on "Verse 41" as golden samples. There are other similar patterns that appear in both, and the overall structure and length are matched.

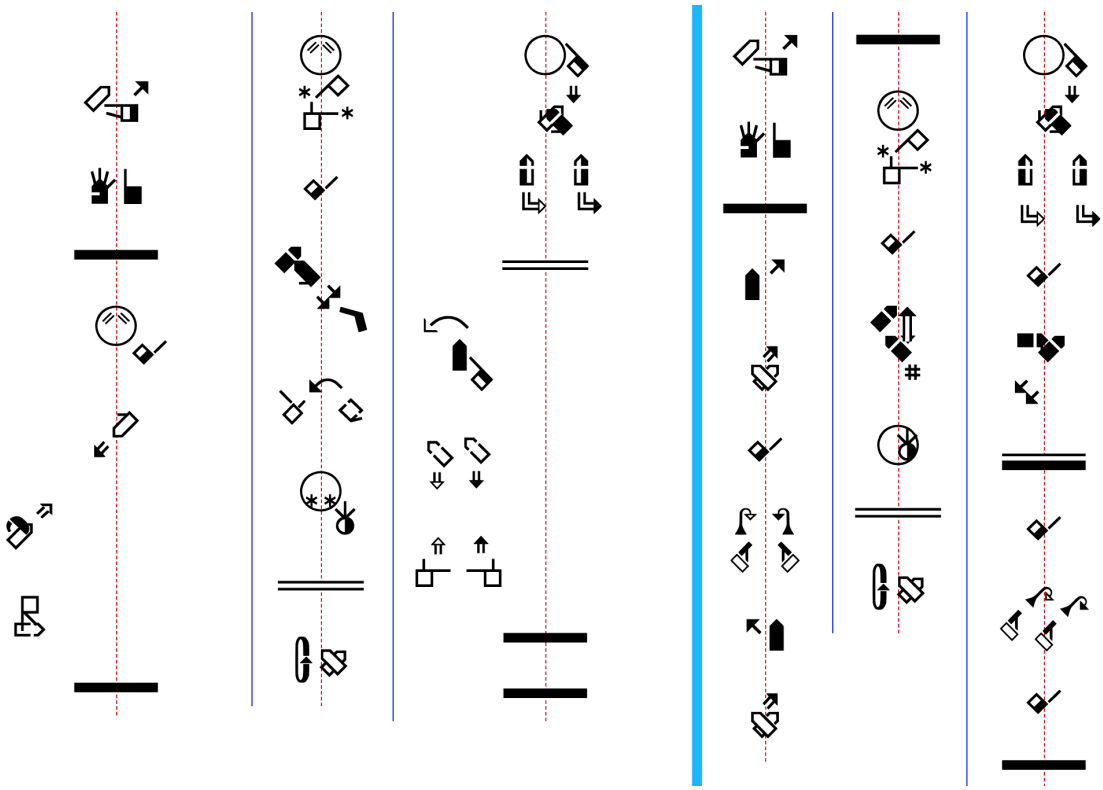


Figure 13: Side-by-side SignWriting evaluation. ASL translation of the English sentence “Verse 41. He gave her his hand and helped her up. Then he called in the widows and all the believers, and he presented her to them alive.” Separated by the vertical light blue (bold) line, on the left is the golden sentence, on the right is the predicted sentence.

5.4 Future Work

There is much possible room for improvement on the experiments of translating between spoken languages and FSW of signed languages.

5.4.1 Fingerspelling Tokenization

As discussed in subsection 5.2.4, the data augmentation method mentioned in subsection 3.8 does not work as expected. Although the current 0.76 accuracy is not very bad, we believe this can be further improved since fingerspelling has a very fixed pattern.

We propose to tackle this problem from the segmentation perspective. As fingerspelling is done letter-by-letter, while BPE segmentation does not tokenize words letter-by-letter, if we can detect fingerspelling in the segmentation/tokenization process, we can then force fingerspelling words to be split letter-by-letter, then it should be trivial for models to learn mappings between fingerspelling signs and Latin letters.

5.4.2 Data Enrichment: Aligning the Bible Corpus

As for data enrichment (discussed in section 3.7 and subsection 5.1.6), instead of using an automatic translation API (DeepL API), a more sophisticated approach is to take advantage of a ready-to-use translation corpus on spoken languages.

At the time of writing, we find a multilingual parallel corpus created from translations of the Bible[1] (Christodoulopoulos and Steedman, 2014), which, if aligned correctly, can be used to translate the $\sim$ 15k American English text in the puddles that contain biblical content (ASL Bible Books NLT and ASL Bible Books Shores Deaf Church, see Table 2), to other 100+ spoken languages.

We try to include the aligned data to the SignBank dataset, which leads to 1000k+ heavily multilingual entries. We believe that a stronger multilingual (on the spoken language side) translation system can be trained on top of that.

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

5.4.3 Regression Objective for Positional Numbers

In our experiments, positional numbers are treated as target factors (see subsection 5.3.1), which contribute cross-entropy loss to the training process. We should not forget that those positional numbers are by nature numeric values, so a regression objective/loss works possibly 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 in Sockeye is only implemented with classification objective (cross-entropy loss). We envision that custom implementation of the regression objective might be useful for our scenario.

5.4.4 Advanced SignWriting Evaluation

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

In our experiments, We separated the evaluation on FSW symbols and positional numbers. For symbols, we borrowed BLEU and chrF from spoken language evaluation, since FSW symbols are the basic graphemes in SignWriting that show many similar linguistic features as spoken language words. For positional numbers, MAE is used, and its limitation is discussed in subsection 5.3.3.

As an improvement, we can use a method similar to teacher forcing (Williams and Zipser, 1989), where we force the output layer of target factors to always predict based on symbol sequences from golden samples. This way, we can eliminate the possible mismatch between predicted and golden symbol sequences.

From a broader perspective, FSW is merely a written notation of SignWriting, which means we can also evaluate on the graphical form, as we did manually in subsection 5.3.5. Moreover, we can exploit computer vision techniques to do automatic comparative evaluation between predicted SignWriting graphics and golden SignWriting graphics.

Ideally, a cascading evaluation method can be applied to SignWriting/FSW - we first evaluate the overall graphics, then on the signs, then on the symbols, then on the positional numbers, and then on the factorized representation of symbols.

6 Conclusion

In this research, we explored building bidirectional and multilingual translation systems between spoken languages and signed languages. Instead of exploiting video representation of signed languages which was done in some past research, we chose to introduce a new research direction of translating between spoken languages and a sign language writing notation system, i.e., SignWriting, to fully make use of the latest advance in MT technology.

SignWriting, despite looking very pictographic, has a well-defined linear writing system called FSW. We created a novel factorization method to factorize FSW into a sequence of symbols and their positional numbers. On the spoken language side, applying common MT preprocessing techniques such as BPE subword segmentation and multilingual tag inclusion proved to be effective in the setting of translating from and to signed languages as well.

We did our experiments based on the SignBank dataset. The first group of models are bilingual, translating from American Sign Language to American English. A factored transformer architecture works, under well-chosen data and hyperparameter combination, as well as some low-resource tricks. Then we extended our experiments to multilingual and to the reverse translation direction, where a novel decoding strategy for FSW is developed.

We borrowed many tools (JoeyNMT, Sockeye) and evaluation metrics (BLEU, chrF) from spoken language translation. On top of them, we added some custom evaluation methods for SignWriting. The performance is good - over 30 BLEU scores in the bilingual setting, and over 20 overall BLEU scores for both translation directions in the multilingual setting. Moreover, similar observations were found as did in some past spoken language translation research, e.g., the curse of multilinguality. All these results confirm the claim that signed languages are full-fledged natural languages, and we should include them in NLP.

Finally, since this research direction is novel, there is much room for improvement, of which we listed some in this thesis. We believe learning more about sign(ed) languages and machine translation would help improve the current systems consid-

erably. We still hope our work to be useful and inspiring, and we feel proud of having accomplished something in a promising new field[1].

[1] Again, sincere thanks to all the people that helped me, without you this work could never have been done.

References

N. M. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. Xydopoulos, K. Antzakas, D. Papazachariou, and P. Daras. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, page 1–1, 2021. doi: 10.1109/tmm.2021.3070438. URL `http://dx.doi.org/10.1109/TMM.2021.3070438`.

R. Aharoni, M. Johnson, and O. Firat. 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, June 2019. doi: 10.18653/v1/N19-1388. URL `https://aclanthology.org/N19-1388`.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*, 2015. URL `http://arxiv.org/abs/1409.0473`.

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

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics, June 2014. doi: 10.3115/v1/W14-3302. URL `https://aclanthology.org/W14-3302`.

O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46. Association for Computational Linguistics, Sept. 2015. doi: 10.18653/v1/W15-3001. URL `https://aclanthology.org/W15-3001`.

M. Borg and K. P. Camilleri. 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, 2019. URL `https://ieeexplore.ieee.org/document/8683257`.

D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M. Ringel Morris. 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, 2019. doi: 10.1145/3308561.3353774. URL `https://doi.org/10.1145/3308561.3353774`.

D. Brentari and C. Padden. 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, 2001.

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

Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):172–186, Jan. 2021. doi: 10.1109/TPAMI.2019.2929257. URL `https://doi.org/10.1109/TPAMI.2019.2929257`.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, June 1996. doi: 10.3115/981863.981904. URL `https://aclanthology.org/P96-1041`.

T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=e42KbIw6Wb`.

Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1212–1221, 2017.

C. Christodoulopoulos and M. Steedman. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49:1–21, 06 2014. doi: 10.1007/s10579-014-9287-y.

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

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

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, June 2005. doi: 10.3115/1219840.1219885. URL `https://aclanthology.org/P05-1045`.

M. Freitag and Y. Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60. Association for Computational Linguistics, Aug. 2017. doi: 10.18653/v1/W17-3207. URL `https://aclanthology.org/W17-3207`.

M. Garcia-Martinez, L. Barrault, and F. Bougares. Factored Neural Machine Translation Architectures. In *International Workshop on Spoken Language Translation (IWSLT'16)*, 2016. URL `https://hal.archives-ouvertes.fr/hal-01433161`.

B. G. Gebre, P. Wittenburg, and T. Heskes. Automatic sign language identification. In *2013 IEEE International Conference on Image Processing*, pages 2626–2630. IEEE, 2013.

R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

T. Hanke, M. Schulder, R. Konrad, and E. Jahn. Extending the Public DGS Corpus in size and depth. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 75–82. European Language Resources Association (ELRA), May 2020. URL `https://www.aclweb.org/anthology/2020.signlang-1.12`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

F. Hieber, T. Domhan, M. Denkowski, and D. Vilar. 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, Nov. 2020. URL `https://aclanthology.org/2020.eamt-1.50`.

W. J. Hutchins. Machine translation: A brief history. In E. Koerner and R. Asher, editors, *Concise History of the Language Sciences*, pages 431–445. Pergamon, 1995. doi: https://doi.org/10.1016/B978-0-08-042580-1.50066-0. URL `https://www.sciencedirect.com/science/article/pii/B9780080425801500660`.

W. J. Hutchins. Machine translation: A concise history. `http://www.hutchinsweb.me.uk/CUHK-2006.pdf`, 2006. Website currently not accessible.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL `https://aclanthology.org/Q17-1024`.

P. Koehn. *Statistical Machine Translation.* Cambridge University Press, 2009.

P. Koehn. *Neural Machine Translation.* Cambridge University Press, 2017.

P. Koehn and H. Hoang. 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, June 2007. URL `https://aclanthology.org/D07-1091`.

J. Kreutzer, J. Bastings, and S. Riezler. 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, Nov. 2019. doi: 10.18653/v1/D19-3019. URL `https://aclanthology.org/D19-3019`.

T. Kudo and J. Richardson. 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: System Demonstrations*, pages 66–71. Association for Computational Linguistics, Nov. 2018. doi: 10.18653/v1/D18-2012. URL `https://aclanthology.org/D18-2012`.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, volume 2, pages 1045–1048, 01 2010.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, Workshop Track Proceedings*, 2013. URL `http://arxiv.org/abs/1301.3781`.

C. D. Monteiro, C. M. Mathew, R. Gutierrez-Osuna, and F. Shipman. Detecting and identifying sign languages through visual features. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 287–290. IEEE, 2016.

A. Moryossef. Sign language datasets. `https://github.com/sign-language-processing/datasets`, 2021.

A. Moryossef and Y. Goldberg. Sign Language Processing. `https://sign-language-processing.github.io/`, 2021.

A. Moryossef, I. Tsochantaridis, R. Y. Aharoni, S. Ebling, and S. Narayanan. Real-time sign-language detection using human pose estimation. In *SLRTP 2020: The Sign Language Recognition, Translation Production Workshop*, 2020.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 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, 7 2002. doi: 10.3115/1073083.1073135. URL `https://doi.org/10.3115/1073083.1073135`.

L. Pishchulin, A. Jain, M. Andriluka, T. Thorm ä hlen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185. IEEE, 2012.

M. Popović. 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, Sept. 2015. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics, Oct. 2018. URL https://www.aclweb.org/anthology/W18-6319.

S. Prillwitz and H. Zienert. 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, 1990.

M. Rikters. *Hybrid Machine Translation by Combining Output From Multiple Machine Translation Systems*. PhD thesis, University of Latvia, 2019.

W. Sandler and D. Lillo-Martin. *Sign language and linguistic universals*. Cambridge University Press, 2006.

P. Santemiz, O. Aran, M. Saraclar, and L. Akarun. 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, 2009.

R. Sennrich. 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, Apr. 2012. URL https://aclanthology.org/E12-1055.

R. Sennrich and B. Zhang. 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, July 2019. doi: 10.18653/v1/P19-1021. URL https://aclanthology.org/P19-1021.

R. Sennrich, B. Haddow, and A. Birch. 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, Aug. 2016a. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

R. Sennrich, B. Haddow, and A. Birch. 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)*, pages

86–96, Aug. 2016b. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

S. Slevinski. Formal SignWriting. Internet-Draft draft-slevinski-formal-signwriting-08, Internet Engineering Task Force, July 2021. URL https://datatracker.ietf.org/doc/html/draft-slevinski-formal-signwriting-08.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

V. Sutton. *Lessons in sign writing*. SignWriting, 1990.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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

R. J. Williams and D. Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, 06 1989. doi: 10.1162/neco.1989.1.2.270. URL https://doi.org/10.1162/neco.1989.1.2.270.

K. Yin, A. Moryossef, J. Hochgesang, Y. Goldberg, and M. Alikhani. 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, Aug. 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.acl-long.570.

B. Zhang, P. Williams, I. Titov, and R. Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.148. URL https://aclanthology.org/2020.acl-main.148.

C. Zhou, D. Levy, X. Li, M. Ghazvininejad, and G. Neubig. 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, Nov. 2021. doi: 10.18653/v1/2021.emnlp-main.458. URL https://aclanthology.org/2021.emnlp-main.458.

Curriculum Vitae

Personal details

Wallisellenstrasse 252
8050 Zürich
(+41) 76 337 6465
zifan.jiang@uzh.ch

Education

since 2019	Master of Informatics at University of Zurich
since 2019	Special Student at ETH Zurich
2011-2015	Bachelor of Software Engineering at Nanjing University

Professional and part-time activities

since 2021	Scientific Programmer (EVOPHON) at University of Neuchâtel
2021	Teaching Assistant for Informatics II at University of Zurich
2018-2019	Front-end Engineer at Wiredcraft
2014-2017	Front-end Intern/Engineer at Zhihu
2013	Front-end Intern at Alibaba

A Tables

language	#samples	#puddles	sentence pairs (>1k)
en-us (American English)	43698	7	✓
en-sg (Singaporean English)	1136	2	
pt-br (Brazilian Portuguese)	42454	3	✓
mt-mt (Brazilian Portuguese)	4118	4	✓
de-de (German German)	24704	3	
de-ch (Swiss German)	4700	2	
fr-ca (Canadian French)	11189	3	
fr-ch (Swiss French)	8806	3	
fr-be (Belgian French)	3439	1	
fr-fr (French French)	1299	2	
es-es (Spanish Spanish)	8806	2	
es-hn (Honduran Spanish)	3399	1	
es-ni (Nicaraguan Spanish)	2150	2	
es-ar (Argentinian Spanish)	1774	2	
ar-tn (Tunisien Arabic)	4965	2	
ca-es (Spanish Catalan)	3419	2	
ko-kr (Korean Korean)	1525	1	
nl-be (Belgian Flemish)	8301	2	
pl-pl (Polish Polish)	2130	2	
sk-sk (Czech Czech)	5780	2	
sl-sl (Slovenian Slovenian)	3808	2	

Table 7: All 21 (spoken) languages involved in this research