Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 4.4: Second distributional intermediate representation based on embeddings - InterL-E

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101017255

Project Information
Project Number: 101017255
Project Title: SignON: Sign Language Translation Mobile Application and Open Communications Framework
Funding Scheme: H2020 ICT-57-2020
Project Start Date: January 1st 2021

Deliverable Information
Title: Second distributional intermediate representation based on embeddings - InterL-E
Work Package: WP 4 – Transfer and InterLingual Representations
Lead beneficiary: University of the Basque Country (UPV/EHU)
Due Date: 30/06/2023
Revision Number: V2.0
Authors: Adrian Nuñez-Marcos, Bram Vanroy, Gorka Labaka
Dissemination Level: Public
Deliverable Type: Other

Overview: This deliverable describes the second version of the Machine Translation Module, the Inter L-E, including the (i) text-to-gloss and the (ii) Sign Language Translation subcomponents.

Revision History

Version #	Implemented by	Revision Date	Description of changes
V0.1	Adrián Núñez-Marcos	17/05/2023	Initial draft
V0.2	Bram Vanroy	25/05/2023	Section 3.2
V1.0	Gorka Labaka	13/06/2023	First internal revision
V2.0	Adrian Nuñez.Marcos	28/06/2023	Incorporate the comments of the partners

The SignON project has received funding from the European Union's Horizon 2020 Programme under Grant Agreement No. 101017255. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the SignON project or the European Commission. The European Commission is not liable for any use that may be made of the information contained therein.

The Members of the SignON Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the SignON Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Approval Procedure

Version #	Deliverable Name	Approved by	Institution	Approval Date
V1.0	D4.4	Shaun O'Boyle	DCU	16/06/2023
V1.0	D4.4	Marco Giovanelli	FINCONS	21/06/2023
V1.0	D4.4	Vincent Vandeghinste	INT	19/06/2023
V1.0	D4.4	Gorka Labaka	UPV/EHU	28/06/2023
V1.0	D4.4	John O'Flaherty	MAC	16/06/2023
V1.0	D4.4	Horacio Saggion	UPF	15/06/2023
V1.0	D4.4	Irene Murtagh	TU Dublin	20/06/2023
V1.0	D4.4	Lorraine Leeson	TCD	16/06/2023
V1.0	D4.4	Mathieu De Coster	UGent	26/06/2023
V1.0	D4.4	Jorn Rijckaert	VGTC	21/06/2023
V1.0	D4.4	Henk van den Heuvel	RU	16/06/2023
V1.0	D4.4	Myriam Vermeerbergen	KU Leuven	17/06/2023
V1.0	D4.4	Rehana Omardeen	EUD	19/06/202x
V1.0	D4.4	Mirella De Sisto Dimitar Shterionov	TiU	26/06/2023

Acronyms

The following table provides definitions for acronyms and terms relevant to this document.

Acronym	Definition
SLR	Sign Language Recognition
MT	Machine Translation
SL	Sign Language
AMR	Abstract Meaning Representation
ASR	Automatic Speech Recognition

Table of Contents

1. Introduction 5

2. Pipeline 5

2.1. Sign Language Recognition module 6

2.2. Machine Translation module 7

2.3. Text-to-gloss module 11

3. Summary & Future Work 13

1. Introduction

This document describes the work carried out in the context of task 4.2 of WP4 “Development of an intermediate representation based on distributional semantics / embeddings (InterL-E)” in the scope of the project “SignON: Sign Language Translation Mobile Application And Open Communication Framework“. The main objective of this project is to provide an affordable application to reduce communication barriers between deaf and hard of hearing people, on the one hand, and mainstream hearing society, on the other.

The system developed for this deliverable builds upon deliverable D4.3 “First distributional intermediate representation based on embeddings - InterL-E”, in which we developed an InterLingual representation that allowed us to encode a message in a spoken language and decode, i.e. translate, it into another spoken language.

We extended the input and output of the InterLingua from supporting only text (both as input and output) to now supporting both SL and text as input and glosses (or text) format as output. The output glosses can be then used to produce an avatar animation of the translated SL utterance as demonstrated in the literature, e.g. [1,2]. Nonetheless, this deliverable does not cover the avatar generation or sign language production/synthesis (see deliverable D5.5 “Second Sign language-specific lexicon and structure”).

This new translation pipeline is divided into the following steps: the Sign Language Recognition (SLR) module, the Machine Translation (MT) module and the AMR-to-Gloss module (where AMR is the acronym used for Abstract Meaning Representation). In Section 2, we will cover step by step each module of the pipeline.

2. Pipeline

The pipeline is divided into three modules: the SLR module, the MT module and the AMR-to-gloss module. See Figure 1 for an illustration.

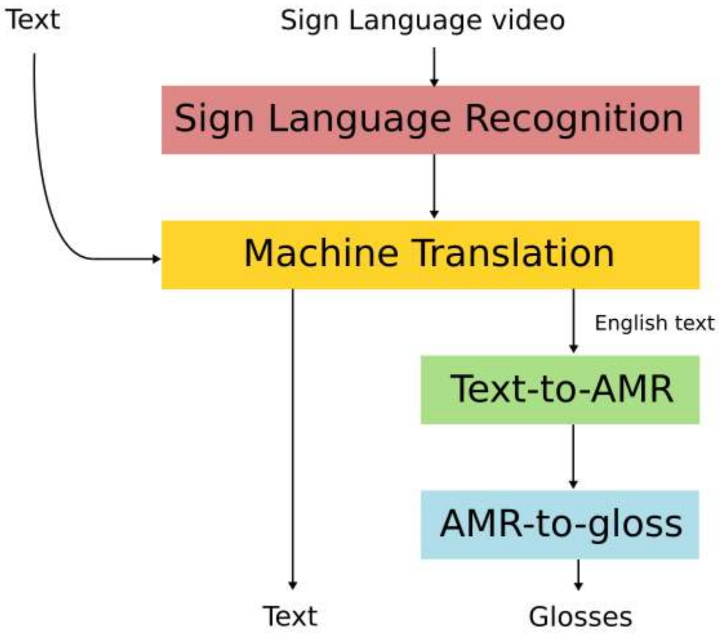


Figure 1. Pipeline overview.

The application can take speech, text or video as input. The speech is processed through the ASR module (see deliverable 3.4 "Automatic Speech Recognition Component and Models") to generate a text transcription. Hence, the MT module described in this deliverable (and in Figure 1) can take two inputs: text (directly written in the app or transcripted from speech) or video. On the one hand, in case text received, in a specific spoken language (Dutch, English, Irish and Spanish), it is directly sent to the MT module. The output format is text again. In case the target output is a SL, the input to the MT module will be translated to English and, then, passed to the text-to-AMR module. The AMR output needs to be fed to the AMR-to-gloss module to transform it to a gloss sequence. On the other hand, when a video is received, it is sent to the SLR module that outputs an intermediate representation that can be used by the MT module to output text. The rest of the process remains the same.

2.1. Sign Language Recognition module

The SLR component takes SL videos and produces an intermediate representation based on embeddings, i.e. a continuous numerical representation of the video. An embedding is a vector composed of real numbers (negative and positives) that contain information about the original video. Each of the embeddings in the output sequence of embeddings has a fixed length of 128 or 512[1] (depending on the SL) and the length of the sequence (number of frames) is relative to the length of the original video. See Figure 2 for a visual example of a sequence of embeddings.

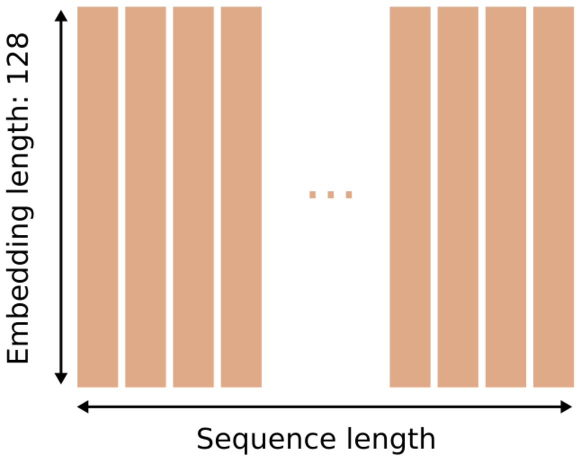


Figure 2. Output embedding sequence.

This output is used as input for the MT module. How the SLR component works is explained in detail in deliverable D3.2 “Sign language recognition component and models”.

2.2. Machine Translation module

The MT module takes (1) raw text or (2) embeddings. The first case is given when the user writes a sentence or speaks an utterance which is converted to text via the ASR module. The resulting text is directly sent to the MT model. In the second case, the embeddings, which represent a SL utterance representation, are produced by the SLR module. Concerning the outputs, the system can output text in the spoken language specified by the user or English text to be used by the text-to-AMR module, depending on the target modality (for SL output, the system will always output English text). The latter’s output is sent to the AMR-to-gloss module.

The text-to-text translation is performed by an mBART [3] neural network. This is already explained in deliverable D4.3 “First distributional intermediate representation based on embeddings - InterL-E”.

[1] Although in deliverable D3.2 “Sign Language Recognition Component and Models” another embedding size is mentioned, we employed an older version (v0.3) of the SLR module. It can be downloaded here: https://github.com/signon-project-wp3/slr-pipeline/blob/main/documentation/inference.md

mBART can translate an utterance from and to one of the following languages: Dutch, English, Irish and Spanish.

The text-to-AMR pipeline was first described in D4.12 “Second adaptable pipeline for training and updating the InterL”. We trained both a multilingual (English, Spanish, Dutch) and English-only model. Because the performance of the English-only model is better than the multilingual model, we decided to work with English as an intermediate lingua franca. That means that a given input text is first automatically translated to English, and this English text is then used as input to the AMR generator. In the coming few months the multilingual AMR model will be revisited and improved so that we do not need to rely on the intermediate translate-to-English step and can solely make use of the multilingual text-to-AMR model.

2.2.1. Embedding-to-text

The embedding-to-text translation is also performed by an mBART model. The original model takes text, extracts subwords and transforms each subword in an embedding that is fed to mBART. We skipped this step and directly sent the embeddings received from the SLR module to mBART. However, as mBART internally requires embeddings of size 1024, we zero padded ours to match that size, i.e. we added zeros until the length of each embedding increased to 1024. Moreover, we subsampled the original embedding sequence as it contained redundancies. We empirically observed that this improved the results. Apart from that, we included a linear layer before feeding the embeddings to mBART (we empirically saw that this improved the results).

For the training, we used the Content4All’s[2] VRT-NEWS dataset (VGT-Dutch). It is composed of videos from news in which a sign language interpreter was interpreting in real time. The training set we used has 5290 samples while the development and test sets have 500 samples (randomly sampled from the dataset).

[2] https://www.cvssp.org/data/c4a-news-corpus/

Figure 3. Content4All's VRT-NEWS dataset sample frames.

We finetuned all the weights (except for the encoder's embedding table, that is not used at all since we use our own input embeddings and not mBART's ones). The training setting was the following: the learning rate was 0.0001 (and reduced its value by monitoring the BLEU-4 on plateaus by 0.7 until a value of 0.0000001 was achieved), the batch size 128 and the weight decay 0.001. We also included an early stopping strategy using BLEU-4. In addition, during the evaluation on the development set, we used a translation beam size of 2 and, in the test set, a beam size of 6. During training, we used a greedy decoding strategy.

With this setting, we obtained the results shown in Table 1 (for the development set after completing the training) and in Table 2 (on the test set).

Table 1. Development set results using the VRT-NEWS dataset.

BLEU-1	BLEU-2	BLEU-3	BLEU-4	CHRF	ROUGE
7.5	1.9	0.56	0.28	18.36	6.85

Table 2. Test set results using the VRT-NEWS dataset.

BLEU-1	BLEU-2	BLEU-3	BLEU-4	CHRF	ROUGE
7.96	2.03	0	0	18.27	7.61

The results obtained by the authors of the dataset in [4] are presented in Table 3, both for development and test set. The results are not directly comparable since they did not share their train/dev/test splits. However, the system developed for this deliverable is comparable to the state-of-the-art model for this dataset. In fact, we can also compare our results with the ones obtained in the First WMT Shared Task on

Sign Language Translation (WMT-SLT22) [6]. Even though the dataset is not the same, the results obtained in terms of BLEU-4 are also lower than 1, more or less comparable to ours.

Table 3. Development and test set results shown in [4] using the VRT-NEWS dataset.

Development		Test	
BLEU-4	ROUGE	BLEU-4	ROUGE
0.45	17.63	0.36	17.77

Moreover, we also performed several experiments in another SL dataset, Phoenix2014T [5], with systems similar to the one we present here (the hyperparameters are adapted in each case to the input features and the dataset, so results can vary). The results were higher in Phoenix2014T, reaching a BLEU-4 of ~22-23 in various experiments. This confirms that the model is correct and can perform much better. Nonetheless, the VRT-NEWS dataset has various difficulties that explain the lower results (see Figure 3 which illustrates the following points):

- The face of the signer is blurred, so a lot of information is lost (non-manual features, for example).
- There is a lot of background clutter since, in contrast to other datasets, the background here contains a person talking and images appearing. Even cropping out the signer, the background still has noise, i.e. it is not monochromatic like Phoenix14T.

In contrast, Phoenix2014T has a smaller vocabulary (~3,000 subwords in comparison to the vocabulary used for VRT-NEWS, which had ~13,000 subwords) as the domain is narrower (weather forecast) and has 1.5 times more training samples. In addition, the videos of this dataset have a monochromatic background, so there is not that much noise in comparison with VRT-NEWS. All this shows that the selected method is at the state of the art level and the low scores obtained in the experiments in VRT-NEWS are due to the limitations of the dataset itself (few and noisy data).

The implementation to train this system can be found in the https://github.com/signon-project-wp4/slt-component repository, while the code used for inference in the application is kept in this repository: https://github.com/signon-project-wp4/embedding2text_translator.

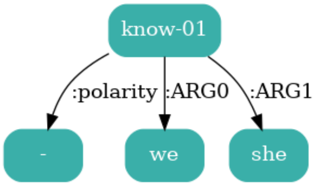
2.3. Text-to-gloss module

To drive the avatar, we need a formal way to send commands to it so that it knows how to move, which handshapes to produce, and the speed in which to behave. Parallel datasets in this regard (text-to-avatar instructions) are scarce. Therefore, we decided to focus on generating glosses instead. To this end we relied on so-called Signbanks for at least VGT, video of signs, their ID glosses, and possible translations into written language. If we can generate glosses, that means that we can also generate sequences of videos (i.e. those that are in the Signbank), although that would be very unnatural. Ideally key-points would be extracted from these videos and smooth transitions generated. Initially the text-to-amr-to-gloss pipeline was developed for VGT (see below). However, gloss-to-SiGML descriptions (Signing Gesture Markup Language) which can be used to drive an avatar was made available for NGT. Therefore, we decided to shift the focus to NGT and develop the complete pipeline using a text-to-amr-to-gloss-to-sigml pipeline.

The pipeline works as follows. First, the input text is translated to English because the AMR model that we trained works best with English input data. Perhaps because the output AMR representation also resembles English text - as shown below. Then the AMR model generates an AMR graph that contains extracted semantic information from the input text (see D4.12 “Second adaptable pipeline for training and updating the InterL” for an example). This intrinsic capability of AMR to move away from the lexical form highlights the strength of using AMR instead of rule-based approaches, as discussed in D4.1 “First symbolic intermediate representation - interL-S”, for glossing. AMR creates an abstract meaning graph about the events and concepts in an utterance, and avoids being literal or tied to surface forms. That means that AMR is a way of reducing the many possibilities of language utterance to a more narrow representation in terms of vocabulary, which is useful if the next step is to look for corresponding glosses in the Signbank, which is a closed vocabulary set. From the AMR graph we can then extract concepts and events, as the cornerstones of the meaning of the sentence as well as useful semantic role types. The extracted concepts are always in English because AMR is based on PropBank semantic frames, which are named in English. But because they denote semantics, they are just a means to define language agnostic events or relations.

As example 1 shows, this means that we can extract core concepts (such as “know”, “we”, “she” for the sentence “We do not know her”) but also additional linguistic information like a negation (“:polarity -” in AMR), which do not necessarily need their own ID gloss in sign language.

Example 1
Text:
We do not know her.
Extracted concepts:
know we she
Extracted “meta”:
negation



The next challenge is to map those English concepts to glosses in the target language (e.g. NGT). One approach would be to use pseudo-glossing through linguistic processing (often related to lemmatization). However, in our case we really want to make sure that the glosses that we generate are from a closed vocabulary that can then be mapped to SiGML. We therefore modify existing SignBanks (applicable to NGT and VGT, but here we focus on NGT), which are dataframes that include for each NGT gloss also a few possible lexical translations in Dutch (see Example 2 for an example).

Example 2

ID:

4092

Gloss:

GEBAREN-B

Dutch translations:

gebaar, gebaren

Video:

https://vlaamsegebarentaal.be/signbank/dictionary/protected_media/glossvideo/GE/GEBAREN-B-4092.mp4

Since the goal is to use English concepts to get NGT glosses, we extend the data representation by providing a separate English column of possible translations. To do so we use multilingual wordnets to find related synsets in English and we use the OpenAI translation API to find in-context lexical translations, which is more feasible to do through a prompting LLM than a traditional MT system.

As a result, the modified dataframe has a column with NGT glosses and their plausible English translations. So given the English extracted concepts from AMR, we can look up the corresponding NGT gloss. As expected, this may lead to ambiguities, that means that some English words are plausible for multiple glosses on a lexical level. To disambiguate we use LABSE vectors to find the gloss option with the highest similarity to the input sentence. LABSE (Language-agnostic BERT Sentence Embeddings; Feng, et al., 2022) is a system to generate vectors that capture the meaning of a given sentence in 109 languages, which allows us to compare sentences (and words) across languages via vector similarity. In our case, if multiple gloss options are possible, we can choose the option that leads to the highest similarity with the English input sentence. Ultimately that leaves us a sequence of NGT glosses, generated from English concepts which in turn were extracted from AMR. Crucially, however, sign order is missing from this pipeline for now. In the future work section we discuss how we are currently looking into more synthetic-data-driven approaches to mitigate such, and other, issues.

The code for the text-to-gloss module as well as preprocessing code for augmenting the Signbanks can be found in this repository: https://github.com/signon-project-wp4/text2gloss.

3. Summary & Future Work

In this deliverable we have explained our current text-to-text, text-to-gloss, SL-to-text and SL-to-gloss pipeline. From the first version in deliverable D4.3 “First distributional intermediate representation based on embeddings - InterL-E”, we have included the possibility of translating from SL and also to translate to glosses which can then be used to generate an avatar animation of the signed utterance.

In terms of gloss generation, we are currently continuing to explore direct text-to-gloss translation via machine translation as already done between Spanish and Spanish Sign Language (LSE) [7]. Aligned corpora of text-to-gloss are rare, so we generate silver corpora of pseudo-glosses. A silver dataset is a large, but noisy, dataset that can be used as a first step of training. It is typically machine-generated and different from “gold” datasets which are verified and/or created by humans. The model trained on the noisy data can then be finetuned on the few gold datasets that we have to improve its performance. This is ongoing research that could scale well to other languages as it is not restricted to whether or not we have access to a SignBank.

As future work, we plan to jointly train the SLR and SLT modules so that both can benefit from each other, i.e. the SLR can benefit from the training objective of the SLT part (outputting a translation) and the SLT from adapting the SLR model to improve the SLT result. Moreover, since the dataset we employed was rather noisy, we will explore more datasets with more controlled backgrounds, e.g. the *Corpus Vlaamse Gebarentaal* (Flemish Sign Language Corpus).

References

[1] Stoll, S., Camgoz, N. C., Hadfield, S., & Bowden, R. (2020). Text2Sign: towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*, *128*(4), 891-908.

[2] Stoll, S., Hadfield, S., & Bowden, R. (2020). Signsynth: Data-driven sign language video generation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16* (pp. 353-370). Springer International Publishing.

[3] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., ... & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726-742.

[4] Camgöz, N. C., Saunders, B., Rochette, G., Giovanelli, M., Inches, G., Nachtrab-Ribback, R., & Bowden, R. (2021, December). Content4all open research sign language translation datasets. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)* (pp. 1-5). IEEE.

[5] Koller, O., Forster, J., & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, *141*, 108-125.

[6] Müller, M., Ebling, S., Avramidis, E., Battisti, A., Berger, M., Bowden, R., ... & Tissi, K. (2022, December). Findings of the first wmt shared task on sign language translation (wmt-slt22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)* (pp. 744-772).

[7] Chiruzzo, L., McGill, E., Egea Gómez, S. & Horacio, H. (2022). Translating Spanish into Spanish Sign Language: Combining Rules and Data-driven Approaches. *Proceedings of the 29th International Conference on Computational Linguistic*, pages 75–83.

[8] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2022). Language-agnostic BERT Sentence Embedding. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 878–891). Association for Computational Linguistics.