Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 4.2: Second symbolic intermediate representation - InterL-S

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 symbolic intermediate representation - InterL-S
Work Package: WP 4 – Transfer and InterLingual Representations
Lead beneficiary: KU Leuven
Due Date: 31/08/2023
Revision Number: Vx.x
Authors: Bram Vanroy
Dissemination Level: Public
Deliverable Type: Other

Overview: This deliverable follows on D4.1 "First symbolic intermediate representation", D4.12 "Second adaptable pipeline for training and updating the InterL", and D4.4 "Second distributional intermediate representation based on embeddings". It describes how a semantic framework, Abstract Meaning Representation, can be used to generate glosses of a restricted vocabulary. Generating glosses allows us to connect the "text" input modality with an avatar through sign language lexica that are being

developed in WP5. This deliverable therefore serves as a bridge between WP4 and WP5 and is currently implemented and running in the SignON app.

Revision History

Version #	Implemented by	Revision Date	Description of changes
v1.0	Bram Vanroy	16/08/2023	Initial version

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.

Approval Procedure

Version #	Deliverable Name	Approved by	Institution	Approval Date
V1.0	D4.2	Shaun O'Boyle	DCU	18/08/2023

V1.0	D4.2	Riccardo Corrias	FINCONS	28/08/2023
V1.0	D4.2	Vincent Vandeghinste	INT	21/08/2023
V1.0	D4.2	Adrián Núñez-Marcos	UPV/EHU	17/08/2023
V1.0	D4.2	John O'Flaherty	MAC	17/08/2023
V1.0	D4.2	Euan McGill	UPF	25/08/2023
V1.0	D4.2	Irene Murtagh	TU Dublin	28/08/2023
V1.0	D4.2	Anthony Ventresque	TCD	24/08/2023
V1.0	D4.2	Karim Dahdah	VRT	23/08/2023
V1.0	D4.2	<Name>	UGent	xx/xx/202x
V1.0	D4.2	Caro Brosens	VGTC	23/08/2023
V1.0	D4.2	Henk van den Heuvel	RU	17/08/2023
V1.0	D4.2	Lien Soetemans	KU Leuven	21/08/2023
V1.0	D4.2	Davy Van Landuyt	EUD	21/08/2023
V1.0	D4.2	Dimitar Shterionov	TiU	17/xx/202x

Acronyms

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

Acronym	Definition
AMR	Abstract Meaning Representation
BML	Behavior Markup Language
LABSE	Language-agnostic BERT Sentence Embeddings
MT	Machine Translation
NGT	Nederlandse Gebarentaal (Sign Language of the Netherlands)
NMT	Neural Machine Translation
SiGML	Signing Gesture Markup Language
SL	Sign Language
VGT	Vlaamse Gebarentaal (Flemish Sign Language)

Table of Contents

1. Introduction

This deliverable describes the text-to-AMR-to-gloss pipeline that is currently implemented in the SignON app as a bridge component between the “text” modality and the avatar for the text-to-sign and sign-to-sign translations, but also covers Abstract Meaning Representation (AMR; Banarescu et al., 2013) as used in SignON more broadly. Over the last months, parts of this pipeline have been described in other deliverables, some very recent, so nothing *new* is added here except for some implementation details. A succinct overview:

- **D4.1** First symbolic intermediate representation - InterL-S (M18)
  First mention of AMR, explanation of similarities between AMR and Sign_A
- **D4.4** Second distributional intermediate representation based on embeddings - InterL-E (M30)
  Explanation of how AMR is used in the text-to-gloss pipeline
- **D4.12** Second adaptable pipeline for training and updating the InterL (M24)
  In-depth explanation of custom-built AMR system, including linearisation and delinearisation
- **D5.5** Second sign language-specific lexicon and structure (M28)
  A discussion of how AMR could be used alongside Sign_A. Either to help predict it or as a separate avenue to work on.

This deliverable serves as a comprehensive overview of the utilisation and development of the text-to-AMR and AMR-to-gloss system, which is currently spread across different deliverables. The experienced reader may have noticed that AMR has been discussed in the InterL-S (symbolic) as well as the InterL-E (embedding) deliverables. Arguments can be made for both. On the one hand, the framework of AMR is symbolic in nature: the linguistic notation system uses a specific, language-agnostic lexicon and structure to describe meaning. On the other hand, the automatic systems that we create are based on neural systems that rely on so-called text embeddings, deep, context-sensitive vectorial representations of language. Such systems are also used in the machine translation (MT) systems described in D4.11 “First adaptable pipeline for training and updating the InterL”. What follows is a summary of how AMR has been discussed in these related deliverables that have touched on AMR, placing particular attention on how we use AMR in SignON as a symbolic interlingua, which can be automatically generated, in the pipeline to go from from spoken language to sign language. First a refresher about abstract meaning representation.

2. Abstract meaning representation

For technical details about AMR the reader is referred to D4.1 “First symbolic intermediate representation” Section 4, or D4.12 “Second adaptable pipeline for training and updating the InterL” Section 2. In short, AMR is an abstract, language-agnostic semantic framework. It moves away from lexical and syntactic form and builds graphs around semantic frames. An example is given below (Example 1) for the sentence “I am so looking forward to October!”, where the event “look forward” has an expressive mode and takes on two required roles, here “I” and October, the tenth month of the year. It also expresses a degree of looking forward, namely “so”.

Example 1: I am so looking forward to October!

Penman:

```
(l / look-forward-03
        :mode expressive
        :ARG0 (i / i)
        :ARG1 (d / date-entity :month 10)
        :degree (s / so))
```

Extracting semantics from surface-level realisations can be useful for a number of applications. For example, having an abstract representation without being bound to lexical or syntactic structures can be beneficial for translation tasks. Indeed, AMR has been used to improve MT systems by providing the NMT model a linguistically unbiased representation of the input (Li & Flanigan, 2022). In SignON, AMR was originally intended as an interlingua to go from text to Sign_A but is currently implemented for text to gloss. Both of these will be described below.

3. AMR-to-Sign_A

Sign_A (Murtagh, 2019) was initially intended as the linguistic framework to drive the avatar (cf. D5.4 “First sign language-specific lexicon and structure” for an extensive overview). D5.5 “Second Sign language-specific lexicon and structure” implemented extended Behavior Markup Language (BML) directives so that Sign_A could be executed by an avatar, theoretically allowing for text-to-Sign_A-to-BML. That deliverable also sheds more light on the challenges that come with Sign_A and its interdependence on Role and Reference Grammar (RRG) and Sign_A, notably how RRG builds semantic logical structures which are syntactically augmented with “signs as words” by Sign_A. However, automatically parsing any given text in any one of the project’s languages to Sign_A+RRG is not available currently. Preliminary research in D4.1 “First symbolic intermediate representation” suggested that some components of AMR could be mapped to Sign_A+RRG (Figure 1), which gave rise to the idea to automatically generate AMR from text and convert that AMR to Sign_A+RRG.

Figure 1. An example of how AMR and the logical structure of Sign_A relate to each other (taken from D4.1).

English Sentence	AMR	Logical Structure
Excuse me	(z0 / excuse-01 :ARG0 (z1 / you) :ARG1 (z2 / i))	[**do’**(1sg, **say’**(1sg, **excuse’**(2sg, 1sg)))]
Thank you	(z0 / thank-01 :ARG1 (z1 / you))	[**do’**(1sg, **say’**(1sg, **thank’**(1sg, 2sg)))]
Do you use SignON?	(z0 / use-01 :ARG0 (z1 / you) :ARG1 (z2 / product :wiki - :name (z3 / name :op1 "Signon")) :polarity (z4 / amr-unknown))	[**be’**(Q-do, **use’**(2sg, SignON)]

Prior work (e.g., Bevilacqua et al., 2021) has shown that text-to-AMR generation is feasible. That would mean that an avatar could be controlled with text via text-to-AMR-to-Sign_A-to-BML or text-to-AMR+Sign_A-to_BML. For a moment, it was also considered to use AMR in combination with key points to merge the semantic properties of AMR with the “syntactic” ones from keypoints (D5.5 “Second Sign language-specific lexicon and structure”; Section 3.2). Unfortunately, while there is overlap in some properties of AMR and Sign_A+RRG, there are also fundamental differences (most importantly that RRG gives a role to syntax whereas AMR explicitly steps away from syntax; also see D5.5 section 2.4). Mapping AMR to Sign_A or merging the two proved unsuccessful, so it has not been possible to implement the (AMR-to-)Sign_A-to-BML pipeline to date. Alternative approaches were looked into instead, such as the gloss-based approach that is discussed in Section 5.

4. Multilingual AMR system

Because of the initial focus on the aforementioned plan to use AMR as an interlingua between text and Sign_A, multilingual AMR systems have been developed. This was discussed in detail in D4.12 "Second adaptable pipeline for training and updating the InterL" so it will only be summarized here.

At the time of writing D4.12, one of the most prominent existing English-to-AMR systems was SPRING (Bevilacqua et al., 2021). In SignON, however, our system should ultimately work for English, Irish, Spanish and Dutch as spoken languages so we created our own multilingual text-to-AMR systems, one English-only system and one multilingual system for Spanish, English and Dutch. Similar to D4.11 "First adaptable pipeline for training and updating the interL" Irish is missing, because we do not train a model from scratch but instead finetune an existing multilingual model on the task of "translating" from a given language to AMR. Specifically, we finetune mBART (mbart-large-cc25), which was not pretrained on Irish so training further on Irish would negatively impact the performance of the whole system. At the time, the plan was to build on the work of D4.11 in a second iteration of the AMR model, considering that there efforts were made to include Irish into mBART. As will be discussed, this has not been necessary due to ability/necessity to rely on English.

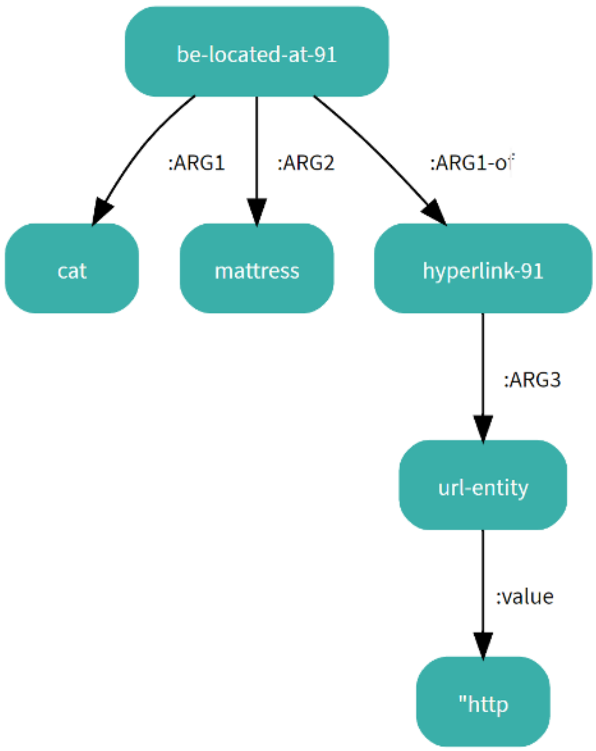
To train the model, a multilingual synthetic dataset was created by means of MT. The English AMR 3.0 Corpus (Knight et al., 2020) was translated with an open-source NLLB model (NLLB Team et al., 2022) to Spanish and Dutch, yielding a dataset of Dutch, Spanish and English to AMR. The training, development, and test sets for each language contain 55,635, 1722, and 1898 sentences respectively. Importantly, only the text is translated but the corresponding AMR stays the same. In other words, one AMR graph in the dataset will correspond to an English, Dutch, and Spanish sentence. AMR is language-agnostic after all.

Because mBART is a sequence-to-sequence model, the AMR hierarchical structure (a graph) needs to be processed into a sequence of tokens so that it can be used by the model, a process called linearisation. To do so, recursive algorithms were created and the original mBART tokenizer adapted so that a graph can be linearised and delinearised back into the original graph non-destructively. This process has been discussed at length in D4.12 "Second adaptable pipeline for training and updating the InterL" and will not be duplicated here.

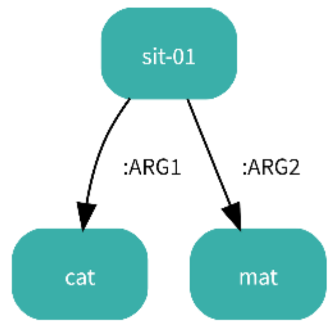
Importantly, however, both the English-only baseline and the multilingual (English, Spanish, Dutch) model achieve good performance on the evaluation set with an accuracy of 0.935 and 0.916 respectively. Both models[1,2] are publicly available and they can be tested out by anyone in a web demo[3]. Upon further usage in downstream tasks (such as AMR-to-gloss, cf. below), it became clear that the multilingual model underperforms for Dutch. For an example see Example 2 below. It was therefore decided to use the MT system developed in D4.11 "First adaptable pipeline for training and updating the InterL" to translate the input text to English as a prior step so that we can always rely on the high-performing English-to-AMR system. While this does add an "error propagation" factor to the pipeline, it is expected that non-English-to-AMR generation is a harder task than general MT into English for the languages that we are working with, and that X->EN->AMR yields better results than X->AMR directly. However, this approach does come with a cost in computation and speed because two steps need to be taken instead of just one. Due to time constraints this approach has only been evaluated superficially and a thorough quantitative analysis has not been done. For future reference it would be worth investigating why the multilingual model does not perform well in practice despite achieving relatively high accuracy on the evaluation set. One explanation might be that it scores well, and perhaps even overfits, on the English part of the dataset but does not perform well on the other languages. Another might be that the quality of the synthetic data is too low, or that AMR is so much like English in terms of its vocabulary that it is an easier task if the input is English. A language-specific evaluation is required to get to the bottom of this discrepancy.

[1] https://huggingface.co/BramVanroy/mbart-en-to-amr
[2] https://huggingface.co/BramVanroy/mbart-en-es-nl-to-amr
[3] https://huggingface.co/spaces/BramVanroy/text-to-amr

Example 2: the difference between using the multilingual system and the English text-to-AMR system.



a. Incorrect output of the multilingual system for the Dutch sentence “De kat zit op de mat”. The model hallucinates an extra “URL” that should not be there.



b. Correct output of the English system for the machine-translated sentence “De kat zit op de mat” -> “The cat is sitting on the mat.”.

With these results in mind, the pipeline as currently implemented in the app always relies on English input text, which can be provided by the high-quality MT system that was developed in D4.11 “First adaptable pipeline for training and updating the InterL”.

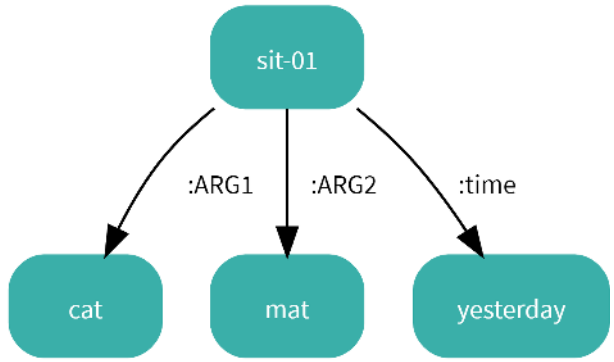
5. AMR-to-gloss

As mentioned in the previous sections, an alternative approach was taken to allow for text-to-signing by an avatar. D5.5 “Second sign language-specific lexicon and structure” describes the use of an NGT-SiGML (Signing Gesture Markup Language) dictionary. Continuous efforts made it possible to map NGT glosses to avatar movements. For specifics of the avatar, see the aforementioned deliverable. With that in mind, the function of the symbolic interlingua had changed: instead of serving as an interlingua between text and Sign_A it now should facilitate the automatic creation of glosses of a closed vocabulary list. The current implementation supports both NGT and VGT, but since the SiGML dictionary is only available in NGT, the remainder of this deliverable will focus on that language.

In the early stages of the project, work was done on the automatic generation of synthetic glosses via linguistic rules such as lemmatization for VGT (cf. D4.1 “First symbolic intermediate representation InterL-S”). NGT was being worked on but not yet finished at the time of writing that deliverable. Unfortunately the core member working on text-to-gloss for these languages left the project shortly after that deliverable. When a new member (the author of this deliverable) joined the team, he started working on AMR, which was the priority then. However, as explained above, the requirements had moved full-circle, all the way back to the need for glosses. Because the new member did not take part in the prior work on text-to-glosses, a different approach was taken, relying on the newly created AMR system. This text-to-gloss system has been described in D4.4 “Second distributional intermediate representation based on embeddings” and is repeated here for completeness’ sake.

SiGML descriptions for NGT are available, as was discussed before. Putting all pieces together, that would yield a text-to-AMR-to-gloss-to-SiGML pipeline for NGT. Retrieving an AMR structure through the automated system has been introduced in an earlier section so the remaining steps are: extracting semantically meaningful concepts from the AMR structure, and finding the compatible gloss in the closed vocabulary. Extracting concepts from AMR is relatively easy. Slightly modifying example 2b (Ex. 3), we can see that the important semantic events and concepts (a sitting event that occurred yesterday, where a cat does the sitting on an object, the mat) are nodes in the graph and the edges are relationships between them. Note that these concepts are always in English because the AMR vocabulary makes use of PropBank semantic frames.

Example 3: AMR structure for “The cat was sitting on the mat yesterday” (extracted concepts sit, cat, mat, yesterday)



5.1. Augmenting SignBanks

Having access to the core semantic concepts, the next challenge is to map these English “words” (concepts) to NGT glosses of a closed vocabulary, i.e., glosses that can in turn be mapped to SiGML. In order to do so, the NGT SignBank has been utilised (the process described here has also been done for VGT’s SignBank). It is a table that contains ID glosses augmented with potential Dutch translations among other data. An additional column was added to the table with potential English translations of a given ID gloss. Two steps were taken to do this: first, context-sensitive translation was used by prompting OpenAI’s gpt-3.5 model (Figure 2). So for a given ID gloss, the prompt could indicate that the related Dutch translations are synonyms, and that gpt-3.5 had to come up with lexical translations that correspond to the *sense* of those given translations. Doing so already takes away some potential ambiguity concerns - which is something that is not easily feasible by only relying on traditional dictionaries or MT engines.

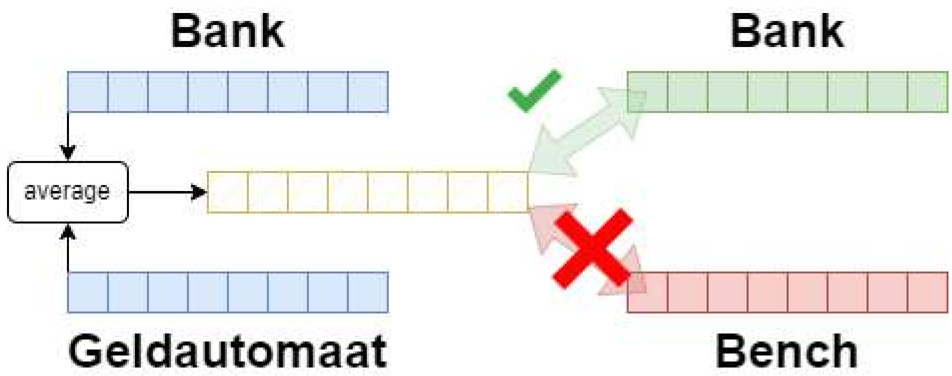
Figure 2. Prompt used for translating with OpenAI's gpt-3.5.

Consider the following list of one or more Dutch words. They are synonyms. Provide one or, preferably, multiple translations of this Dutch 'synonym set' of concepts in English. Format the resulting list of possible translations as a JSON list (without markdown ```json``` marker) and do not add an explanation or any extra information. If you cannot translate a word (such as a city or a name), simply copy the input word.

{List of Dutch words, separated by a comma}

Secondly, Open MultiLingual WordNet (Bond & Paik, 2012) was queried for every Dutch translation, and corresponding English words were added. Unfortunately the result is very noisy: one Dutch word may have many senses, and all the corresponding English words are retrieved, which may not be relevant to the specific meaning that relates to the gloss in question. An alternative WordNet approach was also tested. When multiple Dutch translations were present for a gloss, the English translation corresponding to their first common ancestor (hypernym) was used. This resulted in a vocabulary that was too broad and did not correctly correspond to the glosses. Therefore, the WordNet-translations of the individual words were used instead. To disambiguate and filter out these irrelevant English translations, an additional step was introduced that makes use of multilingual semantic similarity computation. For every English translation that was collected from the gpt-3.5 and the WordNet translation steps, a word embedding was retrieved using LABSE, a state-of-the-art system for language-agnostic embeddings (Feng et al., 2022). A word embedding is a vectorial, latent representation of a word so that similar words have similar vectors. The vector of the English translation candidate was then compared with the average of the vectors of the established Dutch translations. If the vector was similar (cosine similarity of > 0.5), the English translation was retained as a plausible translation of the gloss, otherwise it was removed. An example of the disambiguation process is given in Example 4.

Example 4: (fictional example) Given two Dutch translations of a gloss BANK, "bank" and "geldautomaat" (ATM), an average vectorial representation is created. Potential English translations are compared against this "sense vector" (which would likely imply something to do with money). The English word "bank" is semantically similar enough to that average but "bench" is not so in the final result, “bank” would be included as an English translation of the gloss BANK but “bench” will not.



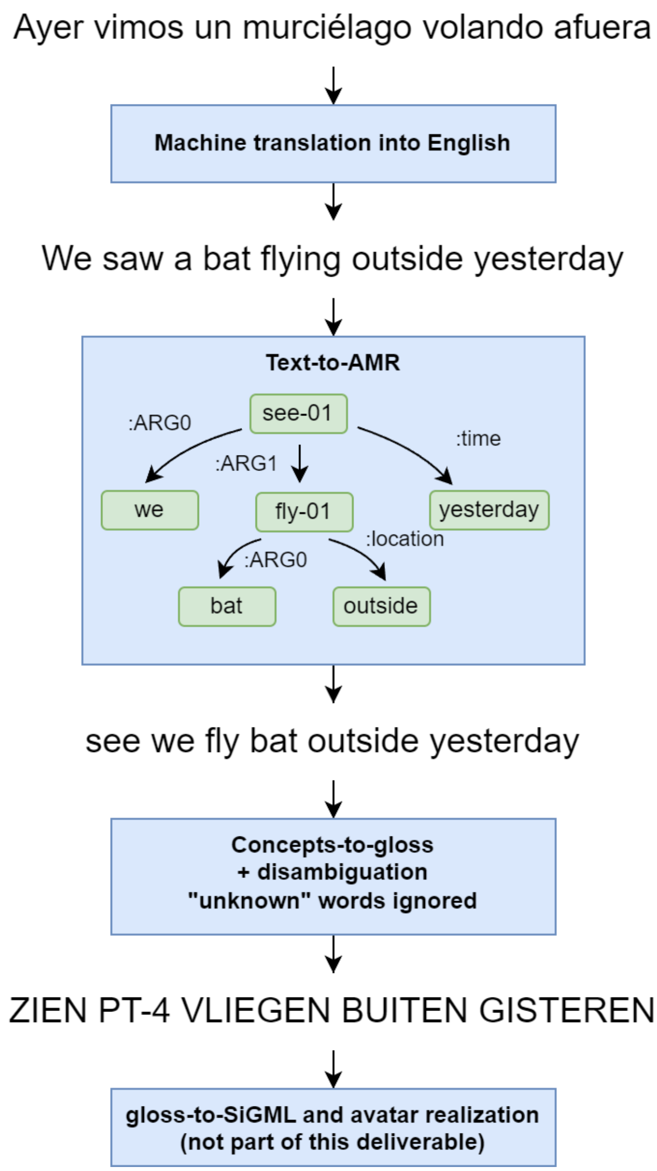
The result of this data processing step is a database that can map the English concepts that we extract from AMR to corresponding NGT glosses and the other way around. As mentioned, this has also been done for VGT because a SignBank is available for that language but because there are no VGT-to-SiGML lexica available on the synthesis side, the VGT pipeline is currently not being used.

5.2. Text-to-gloss pipeline

The text-to-gloss pipeline discussed here is currently implemented in the SignON app and consists of a translation step into English, then a text-to-AMR component so that we can easily extract core concepts and events from the input, and a dictionary mapping between such English concepts and NGT glosses (the same text-to-gloss pipeline was constructed for VGT as well but because there is no VGT-SiGML lexicon, this pipeline is currently not in use). It is possible that an English concept is related to multiple glosses. To disambiguate, we again make use of word embeddings: the different gloss options (lowercased, without regional identifier) are compared with the input sentence (in the input language) and the gloss that yields the highest similarity to the original input sentence is selected. This disambiguation method can likely be improved further by comparing the average vector of the Dutch/English translations of the gloss rather than the gloss itself.

An overview of the pipeline is given in Example 5. It also shows a shortcoming of the current approach as well as a crucial missing component. First, the gloss for “bat” (“vleermuis”) does not exist in the NGT SignBank. It is therefore impossible for this approach to generate it. When a concept does not exist in the SignBank and/or in the NGT-SiGML lexicon (which is called “out of vocabulary”), it cannot be realised by the avatar. It should be noted here that the established lexicon of sign languages is smaller than that of spoken languages due to SL's historical socio-cultural oppression and the impact of being surrounded by spoken languages. Despite these lexical gaps, signers can utilise other strategies to describe certain concepts, which are more flexible in recombining and reusing elements. These strategies are harder to document completely and harder to grasp in a simplified annotation method like glossing (also see Vandeghinste et al., 2023, for more information). Secondly, our pipeline does not include word reordering rules yet so the output order of the glosses is likely incorrect (a naive depth-first order is used based on the AMR graph). Some rules are implemented, based on the pre-review feedback of sign linguists in the consortium, for instance on how to handle names of cities or railway stations, the word order in specific examples, and negation. In agreement with WP5, negation is not explicitly written as a gloss in the pipeline but passed down to the next component as metadata.

Example 5: The pipeline of going from the Spanish sentence "Ayer vimos un murciélago volando afuera" into NGT glosses. The green boxes in AMR indicate the events and concepts that we extract.



The code for this pipeline is available online.[4] It contains instructions on how to run the pipeline. It is also integrated in the workflow in the app.

[4] https://github.com/BramVanroy/text2gloss/

6. Conclusion and current work

In this deliverable, the status, or rather the chronological application, of abstract meaning representation in SignON has been summarised. During the last months (M18-M30), AMR has been discussed in a number of other deliverables, so this deliverable serves as a comprehensive overview with references to related deliverables where applicable. In its latest state, AMR is used to generate glosses. This comes with limitations, however. Glosses are a limited representation of sign language (Yin & Read, 2020), and currently we only have glosses-to-SiGML lexicons for NGT in WP5 to drive the avatar, not for other languages. The continued development of the current approach (using AMR to generate glosses) will depend on upcoming internal meetings to establish the consortium's point of focus. That being said, if the move forward is to continue the gloss-based approach, a number of improvements can be made to the pipeline suggested here.

First, the current disambiguation system generally works well but can be improved further as suggested in the text. Second, a rule-based word-reordering system can be implemented. Some work on this was already done in D4.1 "First symbolic intermediate representation" for VGT and NGT, and VGT sign linguists of VGTC and KU Leuven (Margot Janssens and Lien Soetemans) have very recently finished research on constituency order that will be helpful. In addition, D4.8 "Final routines for transformation of text from and to InterL" describes a method to reorder glosses based on Topic-Comment argument order for BSL, which may serve as inspiration. Third, while not discussed here, Schuurman and colleagues have been working on the conceptualization of SignNets, an alternative to WordNets tailored to sign language (cf. Schuurman et al., 2023). WordNets are semantic frameworks that contain hierarchical relationships between lexical words and their meanings. Collaborations were sought with Francis Bond and colleagues of the Open Multilingual WordNet endeavour, and a meeting was held on how to incorporate and/or develop SignNets as part of a multilingual framework. This work is not ready for use, but theoretical advances can be useful for this project (for instance to improve disambiguation) and sign linguistics in general. Fourth, and most crucial, human evaluation by sign linguists of the output of the systems and the intermediate output is needed. Due to time constraints before the review this has not yet been possible. In addition, considering that there are many moving parts in the current pipeline presented here, it should be worthwhile to compare its results with rule-based glossing that was suggested in D4.11 "First symbolic intermediate representation", although this will require effort of its own considering the core author of that implementation has since left the project. Evaluation should also be done to confirm the aforementioned hypothesis that X->EN->AMR performs better than X->AMR directly.

References

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract Meaning Representation for Sembanking. *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. https://aclanthology.org/W13-2322

Bevilacqua, M., Blloshmi, R., & Navigli, R. (2021). One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(14), Article 14.

Bond, F., & Paik, K. (2012). A Survey of WordNets and their Licenses. *Proceedings of the 6th Global WordNet Conference*. GWC 2012, Matsue.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2022). Language-agnostic BERT Sentence Embedding. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 878–891. https://doi.org/10.18653/v1/2022.acl-long.62

Knight, K., Badarau, B., Baranescu, L., Bonial, C., Griffitt, K., Hermjakob, U., Marcu, D., O'Gorman, T., Palmer, M., Schneider, N., & Bardocz, M. (2020). *Abstract Meaning Representation (AMR) Annotation Release 3.0* (p. 153936 KB) [dataset]. Linguistic Data Consortium. https://doi.org/10.35111/44CY-BP51

Li, C., & Flanigan, J. (2022). Improving Neural Machine Translation with the Abstract Meaning Representation by Combining Graph and Sequence Transformers. *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, 12–21. https://doi.org/10.18653/v1/2022.dlg4nlp-1.2

Murtagh, I. E. (2019). *A Linguistically Motivated Computational Framework for Irish Sign Language* [Thesis, Trinity College Dublin. School of Linguistic Speech & Comm Sci. C.L.C.S.]. http://www.tara.tcd.ie/handle/2262/89131

NLLB Team, Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., … Wang, J. (2022). *No Language Left Behind: Scaling Human-Centered Machine Translation* (arXiv:2207.04672). arXiv. https://doi.org/10.48550/arXiv.2207.04672

Schuurman, I., Declerck, T., Brosens, C., Janssens, M., Vandeghinste, V., & Vanroy, B. (2023). Are there just WordNets or also SignNets? *Proceedings of the 12th Global WordNet Conference*. 12th Global WordNet Conference (GWC 2023). http://hdl.handle.net/1854/LU-01GQ74DP89SP7VAJBAHW7PSDTQ

Vandeghinste, V., De Sisto, M., Kopf, M., Schulder, M., Brosens, C., Soetemans, L., Omardeen, R., Picron, F., Van Landuyt, D., Murtagh, I., Avramidis, E., & De Coster, M. (2023). Report on Europe's Sign Languages. ELE Project Deliverable 1.40. https://zenodo.org/record/8047005

Yin, K., & Read, J. (2020). Better Sign Language Translation with STMC-Transformer. *Proceedings of the 28th International Conference on Computational Linguistics*, 5975–5989. https://doi.org/10.18653/v1/2020.coling-main.525

