Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 3.5: First Natural Language Processing Pipelines

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: First Natural Language Processing Pipelines
Work Package: WP 3 - Source Message Recognition, Analysis and Understanding
Lead beneficiary: Universitat Pompeu Fabra
Due Date: 30/04/2022
Revision Number: V0.3
Authors: Euan McGill, Maud Goddefroy, Horacio Saggion
Dissemination Level: Public
Deliverable Type: Demonstrator

Overview: This deliverable reports on the functionalities of the Natural Language Understanding pipelines developed in the context of the SignON project.

Revision History

Version #	Implemented by	Revision Date	Description of changes
V0.1	Horacio Saggion	12/01/2022	Starting document, table of contents
V0.2	Euan McGill, Horacio Saggion, Maud Goddefroy	05/04/2022	First full draft
V0.3	Euan McGill, Maud Goddefroy	20/04/2022	Address review comments, add info about normalisation, finish section 4

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
V0.2	D3.5	Aoife Brady	DCU	07/04/2022
V0.3	D3.5	Marco Giovanelli	FINCONS	20/04/2022
V0.2	D3.5	Vincent Vandeghinste	INT	11/04/2022
V0.2	D3.5	Gorka Labaka	UPV/EHU	20/04/2022
V0.2	D3.5	John O'Flaherty	MAC	06/04/2022
Vx.x	D3.5	Josep Blat	UPF	07/04/2022
V0.2	D3.5	Irene Murtagh	TU Dublin	25/04/2022
Vx.x	D3.5	Lorraine Leeson	TCD	25/04/2022
Vx.x	D3.5	Mathieu De Coster	UGent	20/04/2022
V0.2	D3.5	Jorn Rijckaert	VGTC	14/04/2022
Vx.x	D3.5	Ellen Rushe	NUID UCD	19/04/2022
Vx.x	D3.5	Henk van den Heuvel	RU	19/04/2022
V0.3	D3.5	Tim Van de Cruys	KU Leuven	22/04/2022
V0.2	D3.5	Davy Van Landuyt	EUD	06/04/2022
Vx.x	D3.5	Mirella De Sisto	TiU	12/04/2022

Acronyms

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

Acronym	Definition
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
Dx.y	SignON Deliverable
E2E	End-to-End
InterL(-E, -S)	Interlingua (embedding, symbolic)
I/O	Input/Output
LSE	Lengua de Signos Española (Spanish Sign Language)
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
OVS	Object-Subject-Verb
PoS	Part of Speech
SL	Sign Language
SLR	Sign Language Recognition
SVO	Subject-Verb-Object
Tx.y	SignON task

VGT	Vlaamse Gebarentaal (Flemish Sign Language)
WP	SignON Work Package
WSD	Word Sense Disambiguation

Table of Contents

1. Introduction

This document reports the current progress in building a project-specific NLP pipeline for SignON, its place in the app infrastructure, and demonstrates use cases for this component. It reports progress made by M16 (April 2022) of the 36 month project timescale. The component takes typed text or text from the ASR component as input, and prepares this source for inclusion in the InterL. The InterL is a language-independent representation of meaning and grammatical function of a given word or utterance, and NLP processes allow this representation to be extracted from text input. This representation may be symbolic, rooted in logical representations for semantics and syntax, or an

embedding for the neural translation model. Figure 1 shows the NLP pipeline as part of the project infrastructure.

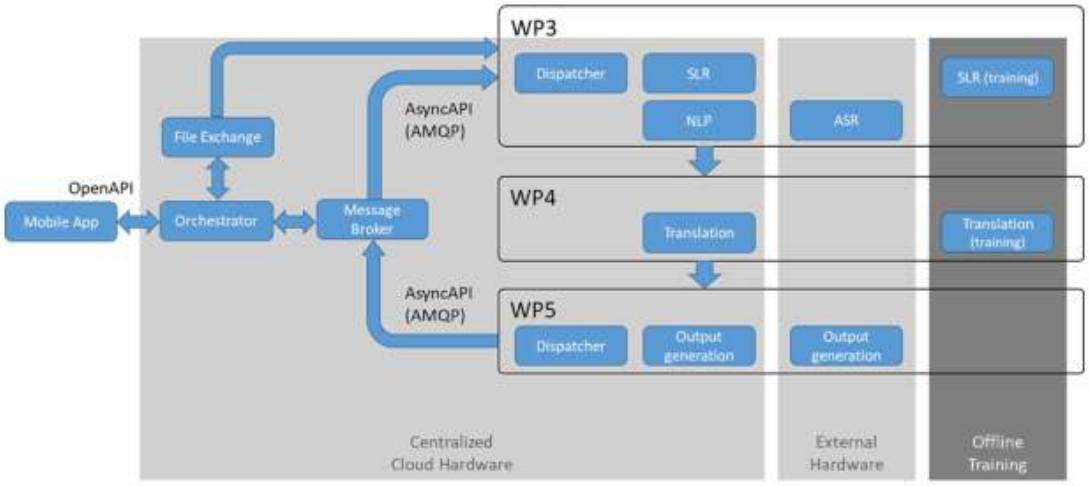

Figure 1: (From SignON D2.3) SignON Framework architecture

Work Package 3 (WP3) task T3.5 generates an InterL representation using NLP techniques to extract semantic and pragmatic meaning, and tokenises and tags text for multimodal translation between sign and spoken languages. The following subsections describe NLP in more detail, particularly tasks which are relevant to SignON. Section 2 describes the NLP models used for each spoken language in SignON, along with additional processing tools, and presents evaluation metrics for each model. Section 3 demonstrates use cases for the NLP module, including using data augmentation techniques and word-sense disambiguation (WSD) to generate training data for low-resource Flemish Sign Language (VGT) from written Dutch. Another use case demonstrates more data augmentation techniques using the NLP module's tagging and parsing capabilities in order to generate Spanish Sign Language (LSE) training data from Spanish text. The final sections analyse the current NLP pipeline module and what is required to fully meet the specifications of the SignON app.

1.1 Text processing and Natural Language Understanding

In order to analyse the contents of a given text by computer for a specific task, it is necessary to break down the utterance into its constituent parts. This consists of understanding features of each individual word. The process of dividing words or subwords into meaningful units is known as *tokenisation* and is required before further analysis is made on each token's grammatical and semantic characteristics.

Thereafter, sequence labelling techniques can be used to assign tags to each token based on its place in the syntactic structure (*dependency parsing*), the semantic role category (*part-of-speech (PoS) tagging*), or the grouping of inflected word forms into the parent lexical grouping (the form found in a dictionary - *lemmatisation*). These tags may be useful themselves for analysis, or fed into other applications or downstream processing tasks. Figure 2 shows these sequence labelling tasks as a pipeline originating from a raw input text. One such example is *named entity recognition (NER)*. Tokens tagged as proper nouns through part of speech tagging are then labelled with a subclass of proper nouns. These classes are not as fixed as grammatical categories mentioned in the sequence labelling tasks PoS-tagging and dependency parsing, but typically include things like people, organisations, geographical locations, and expressions using numbers. Figure 3 shows a text tagged with PoS labels and the proper nouns highlighted for NER analysis.

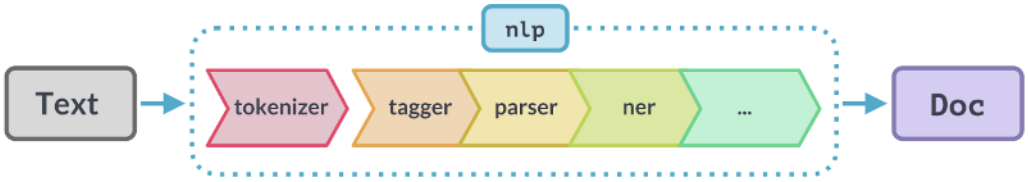

Figure 2: Example of an NLP processing pipeline from input text to processed output[1]

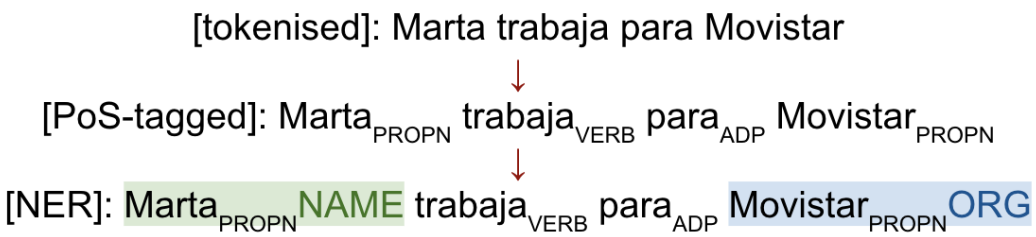

Figure 3: A Spanish text processing pipeline first tokenised, part-of-speech tagged, and then highlighted through named entity recognition. English translation: “Marta works at Movistar”.

Another important process is text normalisation. Entities such as dates and numbers need to be in a text format which is consistent which allows more efficient, accurate, and systematic processing of text data. This process usually occurs before any tagging has taken place.

[1] Image source: https://spacy.io/usage/processing-pipelines

1.2 State of the art

Modern methods in NLP may not require this 'pipelining' approach. Rather, some tasks are achieved by capturing features such as part of speech and other semantic aspects within embeddings. Embeddings are then fed into powerful neural models which operate end-to-end (E2E), being tuned on a particular NLP task. For example, the transformer-based BERT language model (Devlin et al., 2018) can be fine-tuned to have industry-leading performance in question answering, sentiment analysis, and text summarisation among others.

There are some drawbacks in a task-specific neural approach. Firstly, applying fine-tuning for several NLP tasks is resource-intensive, both computationally and in terms of the amount of training data required. Secondly, as SignON aims to build multilingual models which are as self-contained as possible, neural models are not available for all tasks in a given language, or for low-resource languages, such as Irish, at all. Moreover, statistical methods such as those used in sequence labelling pipelines currently have performance approaching state-of-the-art levels in terms of accuracy[2].

2. Preparatory and related work

A wide range of open source libraries for NLP are available to researchers. In order to select one for inclusion in SignON's NLP module, it was necessary to compare the list of required functionalities of the project against the capabilities of a given library, the coverage of these modules on the project's spoken languages (Dutch, English, Irish and Spanish), and their performance. SignON's NLP module must contain text normalisation, spelling correction, sentence identification, tokenisation, PoS tagging, lemmatisation, NER, coreference resolution, entity linking, and figurative language handling - with a preference for statistical and less data-demanding methods in low-resource settings. The aim of including these features is to provide the InterL with a rich representation of each language ready for input into SignON's translation models. It is also necessary that the tools used are open source and permitted for use in a research setting.

[2] https://nlpprogress.com/ is a good reference for performance metrics across NLP tasks and across languages

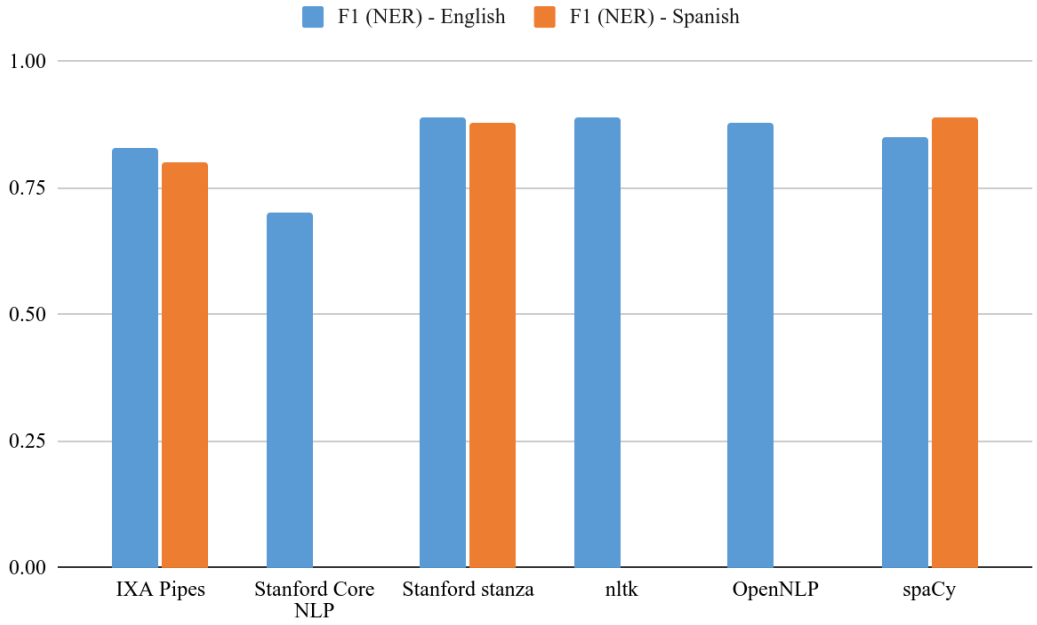

Figure 4: NLP library performance for English and Spanish Named Entity Recognition labelling (F1 score[3]) on the CoNLL-02 dataset

A suitability study was carried out in April 2021, analysing open source NLP libraries against the project requirements. Many libraries such as Freeling[4], GATE[5], IXA Pipes[6], and the Stanford Core NLP[7] libraries were suitable. However, issues which prevented them being chosen include language availability (often lacking support for Irish), models which are not domain-agnostic, lower performance in tagging accuracy metrics, or not being programmable in Python - the preferred language based on the team's knowledge. Figure 4 provides a demonstrative comparison of library performance based on a selected metric: F1 scores for NER accuracy on CoNLL data for English.

The spaCy[8] library was chosen for Dutch, English, and Spanish, while for Irish the Stanford stanza[9] model *UD-Irish-IDT* through the spaCy interface[10] was chosen in order to maintain a framework as unified as

[3] Data for Stanford CoreNLP, OpenNLP and nltk comes from Pinto et al.'s (2016) study. The other data comes from each library's own website
[4] https://nlp.lsi.upc.edu/freeling/node/1
[5] https://gate.ac.uk/
[6] https://ixa2.si.ehu.eus/ixa-pipes/
[7] https://stanfordnlp.github.io/CoreNLP/
[8] https://spacy.io/usage/facts-figures
[9] https://stanfordnlp.github.io/stanza/available_models.html
[10] https://github.com/explosion/spacy-stanza

possible. Table 1 and Table 2 show the functionality and performance of each language's model used in the unified NLP pipeline. All languages' models show adequate functionality for the requirements of the project. With Irish, the morphological information exists in the models, but is not encoded in a consistent way with the other SignON languages. There is also no semantic information through Tok2Vec or similar available, so a third party or novel workaround must be found. As for model performance, Table 2 shows that all models have adequate performance for tagging, with the exception of NER for Dutch, and parsing for Irish lacking somewhat. In future, testing these models with SignON use case text will determine whether the models meet the demands of the project.

Table 1: Summary of model functionality and performance for each SignON locale

		Model			
		UD-Irish-IDT	**en_core_web_md**	**es_core_news_md**	**nl_core_news_md**
Components	**Tagger**	Yes	Yes	Yes	Yes
	Parser	Yes	Yes	Yes	Yes
	SentenceRec	Yes	Yes	Yes	Yes
	Lemmatiser	Yes	Yes	Yes	Yes
	Morphologiser	Other[11]	Yes	Yes	Yes
	AttributeRuler	No	Yes	Yes	Yes
	NER	No	Yes	Yes	Yes
	Tok2Vec	No	Yes	Yes	Yes
Size	**OnDisk**	161MB	55MB	51MB	58MB
	Words	24,000	685,000	500,000	500,000
	Types		20,000	20,000	20,000
	Dimensions		300	300	300
Licence		CC BY-SA 3.0	MIT	GNU GPL 3.0	CC BY-SA 4.0

Outside of the core components outlined in Table 1, 3rd party addons are used to perform word sense disambiguation (WSD), coreference resolution and entity linking. Many of these packages are developed with spaCy models, making it more desirable as the project's choice of library. The following subsections describe their integration into the NLP pipeline.

[11] Morphological information is available for Irish but not following the same labelling procedure as spaCy models

Table 2: Evaluation of model performance for each language

AccuracyEval[12] [13]	Model			
	UD-Irish-IDT	en_core_web_md	es_core_news_md	nl_core_news_md
Token Accuracy	1.00	1.00	1.00	1.00
Tags Accuracy	0.75	0.97	0.96	0.95
Sentence Seg F1	0.96	0.90	0.97	0.87
Unlabeled dependencies (UAS)	0.83	0.92	0.91	0.87
Labelled dependencies (LAS)	0.74	0.90	0.88	0.82
NER-F1		0.85	0.89	0.76
POS Accuracy	0.94		0.98	0.96
Lemma Accuracy	0.92		0.96	0.82
Morph Accuracy			0.98	0.96

2.1 Word sense disambiguation

To effectively represent meaning on the output of the NLP pipeline, the InterL-S, it is essential to perform WSD. Two key approaches to this are a dictionary-based word-sense mapping and by co-occurrence with other words. There exist databases such as WordNet (Miller, 1995) and one-dimensional embeddings based on these senses such as SensEmBERT (Scarlini et al., 2020) to allow the linking of an unambiguous word meaning with the input text. In the case of SensEmBERT, the representation is also multilingual, a marked advantage for the aims of SignON. The approach currently being explored in SignON is able to be integrated into the present NLP pipeline after the tokenisation and tagging stages, and is described in Section 3.2.

[12] Compiled information from spaCy models.

[13] Computed using https://spacy.io/api/scorer#score for the different models.

2.2 Coreference resolution

Dedicated coreference resolution is also included in the NLP pipeline. This comes from the spaCy addon Coreferee for English[14] and a Dutch implementation[15] of the rule-based Stanford Multi-Pass Sieve Coreference System (Lee et al., 2011). For Irish and Spanish, it is possible and realistic to adapt either system in order to have coreference resolution capability in all SignON spoken languages. Again, coreference - referring to re-occurring entities throughout a dialogue - is an important feature to be included in the InterL-S. Explicitly labelling recurring entities and their anaphora in longer utterances means that they can be identified and linked, and then encoded in the symbolic or logical form of the InterL.

3. First NLP pipeline

The NLP pipeline takes its I/O specifications from the SignON Orchestrator, described in detail in D2.3. Specifically, these are the source text and source language in order to process text with the correct model for the language specification. The source text is either supplied by user input on the keyboard or by the ASR module's recognised text. The source language is pre-specified by the user in the application interface. The NLP pipeline module is only used when both the source language and target language parameters in the Orchestrator are spoken languages. For signed languages, the Sign Language Recognition (SLR) component processes the input utterance and outputs this to the InterL. Figure 5 shows a detailed view of the NLP pipeline module's core functionality.

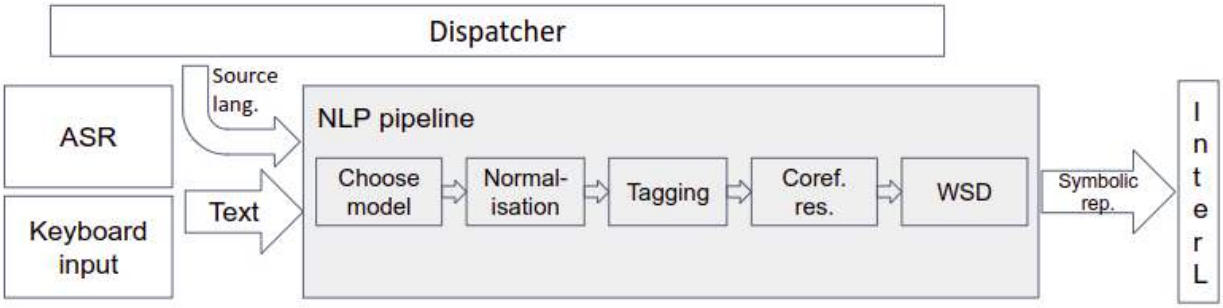

Figure 5: NLP module diagram with surrounding context within the app framework. Relationship with the dispatcher is simplified

[14] https://spacy.io/universe/project/coreferee - a hybrid neural/rule based plugin for spaCy coreference resolution
[15] https://github.com/andreasvc/dutchcoref

Within the pipeline, the chosen spaCy model performs sequence label tagging on the input text. These labels are stored in CoNLL format and it is possible that these could be exported through a format such as JSON for extra output as an additional form of output from the NLP pipeline. Next, the text undergoes a coreference resolution and WSD pass. All of these tags and analyses will provide adequate semantic, syntactic and pragmatic knowledge to provide an underpinning to a symbolic representation of a given input text.

As shown in Figure 1, the NLP pipeline module and its contributors are deeply integrated with other project WPs and partners. Input data from speech relies on T3.4, the ASR module. These speech recognition models are from each SignON spoken language, and there are additional ones tailored to speech from users who are hard-of-hearing and/or wear a cochlear implant. In addition, text from ASR is a stream of tokens without casing and punctuation. Text in this format will require segmentation as part of the normalisation process, and therefore requires cooperation between the ASR and NLU modules within WP3. One possible solution would be to implement a multilingual segmenter[16]. The output specification must adhere to the needs and specifications of WP4, which are currently in development. D4.3 and D4.6 describe the proposed transformation to InterL-E in detail.

The development of this model follows the principle of co-creation exemplified in SignON. This cooperative approach is achieved by hosting the code on SignON's internal Gitlab repository[17]. Here, each change is fully visible by all relevant project partners and each version is approved by a repository maintainer from WP3. The following subsections demonstrate use cases for the NLP pipeline's core functionalities.

3.1 Use case: VGT Data Augmentation

The goal of our text2gloss system[18] is to augment existing corpora that lack gloss-annotations, so that we obtain large enough corpora to train sign2gloss2text models. The rules in the system are based upon linguistic knowledge of the sign language (Flemish Sign Language, VGT in this use case) as much as possible (i.a. Baker et al., 2016; Heyerick et al., 2011; Van Herreweghe, 2011; Vermeerbergen, 2010; Vermeerbergen & Van Herreweghe, 2011). In this deliverable, we focus on how the NLP pipeline is used

[16] Such as https://huggingface.co/oliverguhr
[17] The project's workflow based on version control is discussed in D2.1.
[18] SignON / WP4 / Rule_based_gloss_generation / Rule_based_dutch_to_gloss · GitLab

in the rule-based system and illustrate it with two examples. A more detailed explanation will be included in D4.1.

In the first step, we process each word separately. Here we use lemmatisation to initialise the glosses (cfr. B in examples). Next, we use the POS-tag to filter out all determiners and linking verbs, because these do not have an equivalent in VGT, and punctuation marks because they are typically not used in gloss-annotations. Some other rules, e.g. regarding pointing signs, are activated based on the POS-tag (cfr. C in examples). The heads of each word in the dependency parse are used as well, for example to determine the head of a negation and check whether it is a verb that has a different sign when negated (cfr. D in examples).

In the second step, we sort the glosses, which was done using the dependency labels. We detect the subject (S), verb (V) and object (O) and reorder the sentence to an OVS word order, when the subject is a pronoun (excluding the first person singular) and an SVO word order otherwise (cfr. F in examples). In some verbs, certain parameters can change depending on the subject and/or object. Rules that apply this adaptation, use the dependency labels as well (cfr. E in examples).

Examples:

(A) Je	hebt	het	mij	zelf	gegeven	. (You gave it to me yourself.)
(B) JE	HEBBEN	HET	MIJ	ZELF	GEVEN	.
(C) WG-2	*delete*	WG-3	WG-1	ZELF	GEVEN	*delete*
(E) WG-2		*delete*	*delete*	ZELF	2-GEVEN-1	
(F) ZELF 2-GEVEN-1 WG-2						

(A) Hij	weet	niet	wat	erin	zit	. (He does not know what is in it.)
(B) HIJ	WETEN	NIET	WAT	ERIN	ZITTEN	.
(C) WG-3	WETEN	NIET	WAT	IN	*delete*	*delete*
(D) WG-3	WETEN-NIET		WAT	IN		
(F) WETEN-NIET WG-3 WAT IN						

3.2 Use case: AMuSE-based Word Sense Disambiguation

AMuSE is a ready-to-use WSD tool distributed as a docker image (Orlando et al., 2021). The original tool includes a preprocessing pipeline performing tokenisation, lemmatisation and POS-tagging. We extracted the WSD module, including the trained model from the docker image and integrated those components

in our NLU-pipeline. In future research, we can connect the WordNet synsets, the output of the WSD, to SignNet synsets to achieve better text2sign translations.

Once the contents of D4.1 (“Development of a symbolic intermediate representation”) are published in M18 (June 2022), the NLP pipeline will be able to output a language-agnostic symbolic or logical representation of an utterance’s meaning ready to be fed to the translation models within SignON. The consequence of this is that we will then be able to generate an app-specific InterL form from the pipeline - the most crucial use case in terms of contributing to the SignON platform’s overall functionality.

3.3 Use case: LSE Data Augmentation

Another text2gloss use case using the NLP pipeline’s functionality is for LSE glosses generated from Spanish language data. Here, we use knowledge of LSE grammar (Herrero-Blanco, 2009; Herrero-Blanco, 2010; Morales-López, 2012; Rodríguez González, 1992; San-Segundo et al., 2008), and the conventions of SL glossing, to formulate transformation rules between Spanish text and pseudo-LSE glosses. Table 3 demonstrates some of the rules implemented in the first version of the LSE rule-based data augmenter which utilise the functionalities of the pipeline.. As an example, in (3.3.1) we show an example of LSE’s requirement to specify the subject pronoun where Spanish has no such requirement - as well as the glossing convention where all glosses are typed uppercase.

(3.3.1)

Spanish: tienes que ir a una comisaría →

LSE gloss: **TÚ** NECESITAR IR COMISARÍA

*English: **you** need to go to a police station*

Several aspects of the pipeline are useful in formulating and executing this transformation rule. spaCy’s POS tagger and morphologiser are used to identify the verb in the original Spanish phrase, and then the person and number inflection are identified before the suitable pronoun is inserted before the verb form. (3.3.1) also shows other transformation rules between Spanish-LSE including omission of (in-)definite articles “una comisaría” (*“a police station”*) → “COMISARÍA” (“police station”), omission of prepositions “ir a” (*“to go to”*) → “IR” (“go”), and a glossing preference for “NECESITAR” rather than the compound verb “tener que” (*“to need to”*).

These transformations will be useful in generating more augmented training data for LSE which is *extremely* low resource in terms of corpora available to do machine translation research.

Table 3: Demonstration of rules implemented during Spanish-LSE data augmentation and methods of their implementation using spaCy tools from the NLP pipeline module

LSE grammar rule	spaCy implementation	Source
Derivation: (One LSE gloss reflects all inflected and derived morphological forms)	Lemmatiser	Herrero Blanco, 2009
Copula deletion: Ser and Estar (*"To be"*) are not glossed	Lemmatiser, then find+delete 'SER' and 'ESTAR'	Herrero Blanco, 2009
Noun phrase order: Noun, Demonstrative, Possessive, Numeral, Indefinite	PoS-tagger and dependency parser	Herrero Blanco, 2009
Compulsory subject pronouns	Morphologiser	San Segundo, 2008
Possessive marking with "PROPIO"	Dependency parser, then insert 'PROPIO' gloss	Rodriguez González, 1992

4. Analysis and evaluation of the first NLP pipelines

It is not yet possible to evaluate the planned output functionality of the NLP pipeline, as the specification of the InterL-S has not yet been described. However, it is possible and useful to test the capabilities of the current[19] pipeline iteration. Test phrases could be based on the four main use cases outlined in the grant agreement and provided by partners in WP1. It is hoped that the NLP pipeline will be evaluated in this manner before the next deliverable in December 2023 (M36).

In terms of the requirements of the project, good progress has been made towards a suitable NLP pipeline to be integrated into the SignON platform. All four spoken languages included in SignON have tagging capabilities with a unified infrastructure, with the exception of NER being missing for Irish due to its absence within the Universal Dependencies-based model. There is currently no text normalisation function in the pipeline pending discussion about handling different input specifications (speech versus text) and with partners in the ASR component. Coreference resolution is currently being developed for Dutch and English, and there is a plan in place to include Spanish. There are also resources available for

[19] As of 20th April 2022

WSD. These are crucial for the putative InterL-S specification which are necessary as an input to the SignON translation models, and have been explored in the use cases sections as well as being available in the tok2vec modules of the Dutch, English and Spanish spaCy models. Figurative language handling was another specified requirement. It will not have its own dedicated functionality within the project, as tasks like parsing and WSD handle this type of utterance indirectly.

5. Conclusions and future work

This document demonstrates the NLP pipeline's internal processes, describes its input and output configuration, and contextualises its position within the greater project infrastructure. It also describes possible uses of the internal components of the NLP pipeline, and its connection with other modules and WPs within SignON. Importantly, this report highlights the further work that needs to be completed before the next NLP pipelines deliverable due in M36 (December 2023) which also marks the completion date of the SignON project lifecycle.

Bibliography

Baker, A., van den Bogaerde, B., Pfau, R., & Schermer, G. M. (2016). *The Linguistics of Sign Languages: An Introduction*. John Benjamins Publishing Company. https://books.google.be/books?id=kC01jwEACAAJ

Devlin, J. et al. (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805v2

Herrero-Blanco, Á. (2009) *Gramática Didáctica de la Lengua de Signos Española (LSE).* Fundación CNSE & SM. Madrid, Spain.

Herrero-Blanco, Á. (2010) The Expression of Modality in Spanish Sign Language. In (eds.) Wanders, G. & Keizer, E., *Special Issue: The London Papers II. Web Papers in Functional Discourse Grammar,* 83, 19-42.

Heyerick, I., Van Braeckeveld, M., Rijckaert, J., De Weerdt, D., Van Herreweghe, M., & Vermeerbergen, M. (2011). *Meervoud in Vlaamse Gebarentaal Onderzoeksrapport*. Vlaams GebarentaalCentrum. https://www.vgtc.be/wp-content/uploads/2020/02/2011_meervoud_in_vgt-1.pdf

Miller, George A. (1995) WordNet: A Lexical Database for English. *Communications of the ACM,* 38 (11), 39-41.

Morales-López, E. (2012) Word order and informative functions (topic and focus) in Spanish Signed Language utterances. *Journal of Pragmatics,* 44, 474-489. http://hdl.handle.net/2183/9052

Orlando, R., Conia, S., Brignone, F., Cecconi, F., & Navigli, R. (2021). AMuSE-WSD: An All-in-one Multilingual System for Easy Word Sense Disambiguation. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 298–307. https://doi.org/10.18653/v1/2021.emnlp-demo.34

Pinto, A., Gonçalo Olivera, H. & Oliviera Alves, A. (2016) Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text. In *Proceedings of the 5th Symposium on Languages, Applications and Technologies (SLATE'16).* Maribor, Slovenia. http://dx.doi.org/10.4230/OASIcs.SLATE.2016.3

Rodríguez González, M. Á. (1992) *Lenguaje de Signos.* ONCE & CNSE. Madrid, Spain.

San-Segundo, R. et al. (2008) Speech to sign language translation system for Spanish. *Speech Communication.* 50:11-12, 1009-1020. https://doi.org/10.1016/j.specom.2008.02.001

Scarlini, B., Pasini, T. & Navigli, R. (2020) SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020).* New York, New York, USA. https://doi.org/10.1609/aaai.v34i05.6402

Van Herreweghe, M. (2011). Negatie in de Vlaamse Gebarentaal. *Handelingen - Koninklijke Zuid-Nederlandse Maatschappij voor Taal- en Letterkunde en Geschiedenis*, *54*, 83–108. https://doi.org/10.21825/kzm.v54i0.17262

Vermeerbergen, M. (2010). *Onderzoeksrapport: "Woordvolgorde" in de Vlaamse Gebarentaal*. Vlaams GebarentaalCentrum. https://www.vgtc.be/wp-content/uploads/2020/02/onderzoeksrapport_woordvolgorde-in-vgt.pdf

Vermeerbergen, M., & Van Herreweghe, M. (2011). *Interrogatie in Vlaamse Gebarentaal*. Vlaams GebarentaalCentrum.
https://www.vgtc.be/wp-content/uploads/2020/02/rapport_interrogatie-1.2.2011-1.pdf

Further Reading

General NLP:

Jurafsky, D. and Martin, J. H. (2021) *Speech and Language Processing, 3rd Ed. Draft* Accessed 2022/03/22 at https://web.stanford.edu/~jurafsky/slp3/ (Notable chapters: 6, 8, 11, 14)