Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 3.6: Second Natural Language Processing pipeline

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 Natural Language Processing pipeline
Work Package: WP3
Lead beneficiary: UPF
Due Date: 31/12/2023
Revision Number: V0.2
Authors: Santiago Egea Gómez, Euan McGill, Horacio Saggion
Dissemination Level: Public
Deliverable Type: Demonstrator

Overview: In this document we present and describe the implementation of the second SignON natural language processing pipeline. The pipeline integrates three modules: (1) TextNormaliser, (2) LinguisticTagger and (3) Word-Sense-Disambiguation module.

Revision History

Version #	Implemented by	Revision Date	Description of changes
V0.1	Santiago Egea Gómez	13/11/2023	First draft
V0.2	Santiago Egea Gómez, Euan McGill	30/11/2023	Pre-revision 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
V0.2	D3.6	Shaun O'Boyle, Aoife Brady	DCU	07/12/2023

V0.2	D3.6	Marco Giovanelli	FINCONS	07/12/2023
Vx.x	D3.6	Vincent Vandeghinste	INT	06/12/2023
V0.2	D3.6	Adrián Núñez-Marcos	UPV/EHU	01/12/2023
V0.2	D3.6	John O'Flaherty	MAC	01/12/2023
V0.2	D3.6	Josep Blat	UPF	02/12/2023
V0.2	D3.6	Irene Murtagh	TU Dublin	05/12/2023
V0.2	D3.6	Ellen Rushe	TCD	08/12/2023
V0.2	D3.6	Jorn Rijckaert	VGTC	01/12/2023
V0.2	D3.6	Henk van den Heuvel	RU	08/12/2023
V0.2	D3.6	Catia Cucchiarini	TaalUnie (NTU)	12/12/2023
V0.2	D.3.6	Lien Soetemans Myriam Vermeerbergen	KU Leuven	05/12/2023 10.12.2023
V0.2	D3.6	Davy Van Landuyt	EUD	01/12/2023
V0.2	D3.6	Mirella De Sisto	TiU	06/12/2023

Acronyms

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

Acronym	Definition
G2T	Gloss-to-Text
NLU	Natural Language Understanding
NLP	Natural Language Processing
NER	Name Entity Recognition
PoS	Part-of-Speech
SL	Sign Language
SLT	Sign Language (Machine) Translation
T2G	Text-to-Gloss
WSD	Word-Sense Disambiguation

Table of Contents

1. Overview

This deliverable describes the latest updates in the context of the task T3.5 “Implementing language-specific NLU pipelines” in work package WP3 “Source message recognition, analysis and understanding” of the SignON Project. This task has as its main objective the implementation of a Natural Language Processing (NLP) pipeline to upgrade the information available and used to generate the InterL representation from the input.

The proposed implementation takes text as input and performs the following processes:

- **Text Normalisation**. Users tend to make errors when inputting their message. These typing mistakes can affect the performance of subsequent modules and may alter the source message meaning. Through this process we seek to minimise these errors and normalise the input text.
- **Linguistic Tagging.** It has been shown that lexical, syntactic and semantic information can be used to enhance some NLP models, such as Machine Translation (Egea Gómez et al., 2021; 2022; Chiruzzo et al., 2022; McGill et al., 2023) in a ‘Factored Transformer’ approach (Sennrich and Haddow, 2016; Armegnol Estapé and Costa-jussà, 2021). In the case of Sign Language Translation (SLT), Name Entity Recognition (NER) could also provide relevant information about places and proper names. Additionally, we explored Part-of-Speech (PoS), Morphological and Word

Dependencies as possible features to enhance input representations to translation models. This module is an updated version of the pipeline presented in D3.5.

- **Word Sense Disambiguation (WSD)**. WSD is an important task for NLU. A natural characteristic of language is that one lexeme can correspond to different semantic fields with different meanings. Disambiguating meaning plays a crucial role for success in translation and text processing.

This document is the second of the two describing the SignON NLU pipeline. The first was named "D3.5 - First Natural Language Processing Pipeline" and was published in December 2021. The first deliverable described the progress that had been made so far and its interaction with the other modules within the SignON ecosystem. At that stage, processes for tokenising, tagging, and parsing text data from all four spoken languages of the project (Dutch, English, Irish, and Spanish) under one unified architecture had been constructed - as well as initial experiments on rule-based Word-Sense Disambiguation for Dutch. It was initially planned for the NLU output to be fed to the symbolic representation in the interlingua (InterL-S). However since text is directly passed to AMR during translation, the NLU pipelines will only serve the purpose of tokenising, tagging and normalising text for all spoken languages of SignON - as well as WSD for Dutch, English, and Spanish.

In the rest of the document, we describe the different modules implemented inside the SignON NLU pipeline. Additionally, we have run tests to validate the NLU processes using samples from HoReCo[1] for the project's spoken languages. All the outputs of these tests are presented in the Annex 1 and discussed throughout the document.

2. Text Normalisation

When using the SignON mobile application, users are liable to make mistakes when typing their message in the application via text. These errors can propagate through the translation pipeline and sequentially increase when passing through the different processes. Additionally, speech recognition may not produce a perfect transcription of the message and errors in this phase will also propagate through the whole SignON pipeline.

[1] https://european-language-equality.eu/wp-content/uploads/2023/04/ELE2_Project_Report_NGT_HoReCo.pdf

Through text normalisation, unnormalised text is corrected and standardised before being inputted into the subsequent processing modules. This text normalisation process involves the following two steps:

(1) **Text Normalisation.** This process consists of removing repeating punctuation and prohibited, strange symbols, such as @ or ^. For this purpose, We employed the series of Regex[2] rules presented in Annex 2.

(2) **Spell checking.** This process assures that the messages input to the next modules in the SignON process contain the minimum typing errors possible. Hunspell[3] is one of the most popular spell checking libraries, used in well-known software such as LibreOffice, Mozilla and Google Chrome. Hunspell works at word-level and is implemented in C++, which makes it very efficient. Additionally, it offers an easy way to include different dictionaries allowing us to include all the languages covered in the project. Finally, there is a wrapper for Python which suits our development requirements. Table 1 presents the open source dictionaries used for each language.

Table 1. HunSpell dictionaries used for spell checking

Language	Source
Dutch	https://github.com/OpenTaal/opentaal-hunspell
Spanish	https://github.com/elastic/hunspell/tree/master/dicts/es_ES
Irish	https://github.com/wooorm/dictionaries/tree/main/dictionaries/ga
English	https://github.com/elastic/hunspell/tree/master/dicts/en_GB

The processes described here are implemented in the class TextNormalizer in a GitHub repository[4]. In Table 2, we provide some examples of the TextNormalizer outputs for the different languages.

Discussion & limitations. As is apparent from Table 2, spell checking is not enough to correct some typos, since it is performed without sentence context. The clearest example is in the second English example, in which the typo “spand” has been replaced by “stand” instead of “spend”. As a positive outcome, TextNormalizer is able to correct strange and repetitive punctuations. The worst results were obtained for the Irish. In this case, the hunspell dictionary does not suggest the best replacements for

[2] https://docs.python.org/3/library/re.html
[3] https://github.com/hunspell/hunspell
[4] https://github.com/signon-project-wp3/WP3-Second-NLP-Pipeline/tree/main/TextNormalizer

the typos and some suggestions are at risk of changing the message meaning. In future releases of the pipeline, it would be a positive development to combine grammar checking, which takes into account sentence contexts, with the current approach.

Table 2. TextNormalizer outputs. "R" denotes the original sentence from HoReCo, "I" the altered version used as input and "N" the normalised output. Errors and successes are marked in red and green respectively

ENGLISH **R:** In town to Christmas shop and spend time with family. **I:** In town to Christmass shop and spand time with family. **N:** In town to Christmas shop and **stand** time with family . **R:** Awesome! I will come back! **I:** Awesume!!!!!! I will come back ! **N:** Awesome ! I will come back !
IRISH **R:** Iontach ! Tiocfaidh mé ar ais! **I:** Iontach!!!!!! Tiucfaidh mé ar ais! **N:** Iontach ! **Faitidh** mé ar ais ! **R:** Tá mé ar an mbaile chun siopadóireacht a dhéanamh don Nollaig, agus am a chaitheamh leis an teaghlach. **I:** Ta me ar an mbaile chun siopadóireacht a dhéanamh don Nollaig, agus am a chaitheamh leis an teaghlach. **N:** **Na de** ar an mbaile chun siopadóireacht a dhéanamh don Nollaig , agus am a chaitheamh leis an teaghlach .
SPANISH **R:** ¡Increíble! ¡Volveré! **I:** ¡Increíble! ¡Bolveré! **N:** ¡ Increíble ! ¡ Volveré ! **R:** En la ciudad para hacer compras navideñas y estar en familia. **I:** En la ciudad para hacer compras navidenas y estar en famillia. **N:** En la ciudad para hacer compras navideñas y estar en familia .
DUTCH **R:** Geweldig! Ik kom zeker terug! **I:** Geweldig!!!!!! Ik kom zekerr terug! **N:** Geweldig ! Ik kom zeker terug ! **R:** We waren in de stad om kerstinkopen te doen en tijd door te brengen met familie **I:** We waren in de stud om kerstinkopen te doen en tijd door te brengen met familia **N:** We waren in de stud om kerstinkopen te doen en tijd door te brengen met familie

3. Natural Language Understanding

In this module, the input text is parsed to linguistically enrich the source information. As is shown in the previous deliverable D4.7 “Second Routines for Transformation of Text from and to InterL” and some of our publications (Egea Gómez et al., 2021; 2022; Chiruzzo et al., 2022; McGill et al., 2023), linguistic features can enhance the translation process at least at Spoken-To-Gloss level. Additionally, PoS information is very important for the subsequent NLU process (such as WSD). Although the first version of the SignON service will not use these features for the translation process, it could be very useful in future project stages.

3.1 Parsing of Spoken Languages

For spoken languages, there are well-established models and techniques for linguistic tagging. As Deep Learning approaches have shown very good performance in this task, we leverage the SpaCy library[5] for this task. The implementation described here is an upgrade of the previous version reported in the project deliverable D3.5 “First Natural Language Processing Pipelines”, and can be found in detail in Section 5.

In the rest of this section, we describe other tasks which may be useful to the aims of SignON while using the capabilities of the NLU pipeline, and NLP tasks in general. First, in Section 3.2, we describe how it is possible to use features of the pipeline to generate synthetic glosses to augment the amount of BSL glosses available. Then, in Section 3.3 we discuss possible methods to PoS-tag sequences of SL glosses.

3.2 Using NLP pipelines to generate synthetic BSL glosses

Recently, we have conducted a series of experiments following the data augmentation strategy laid out by Moryossef and colleagues (2021) whereby synthetic glosses are generated in a rule-based manner from a monolingual corpus[6]. This allows us (*e.g.* Chiruzzo et al., 2022; McGill et al., 2023) to pretrain NMT models on a larger base of parallel text and SL gloss data, before fine-tuning on real-world parallel spoken language and SL corpora.

[5] https://spacy.io

[6] The monolingual corpus must be written in the same language that the ID-Glosses are derived from e.g. English for American Sign Language (synthetic) glosses

Table 3. Exemplar rules used to generate synthetic BSL glosses, and the tools used to formulate them

Rule	Example	Tool
Lemmatise all words	Sam went to the pub →SAM GO PUB	NLU Pipeline: SpaCy (English model) part-of-speech tagger
Disallow adpositions, determiners and punctuation		
Reorder NEG+VERB to VERB+NEG	Sheila [does] not own a horse → SHEILA OWN NOT HORSE	NLU Pipeline: SpaCy (English model) part-of-speech tagger and dependency parser
Pronoun+BE+Adj -> Pronoun+Adj+Pronoun	They are polite → THEY POLITE THEY	
Constituent order based on semantics: Time-Location-Object-Subject -Verb-Location	Why was the black cat climbing the tree in your garden yesterday → YESTERDAY GARDEN PT:POSS.2SG TREE CAT BLACK CLIMB WHAT-FOR	AllenNLP (English model) semantic role labelling model
Morphologically-complex glosses for time expressions	Five o'clock → O'CLOCK-FIVE	RegEx

The data problem is particularly acute for British Sign Language (BSL) where the SignON project only has access to less than 1,000 parallel English/BSL utterances. Following the grammar of BSL (*e.g.* Sutton-Spence and Woll, 1999), we implemented a number of lexical, syntactic, and semantic rules in order to generate pseudo-BSL synthetic glosses. These rules are shown in Table 3. We were able to use the functionality of the SignON NLU pipeline for many of these rules, while others were implemented using the AllenNLP toolkit and its semantic role labelling model; as well as some rules being implemented solely with regular expressions.

A more in-depth discussion of this implementation is included in "D4.8 - Final Routines for transformation of text from and to InterL", as well as its impact in conducting *extremely* low-resource translation between text and BSL glosses.

3.3 Part-of-speech tagging sign language gloss data

While running experiments for LSE→Spanish G2T translation (*c.f.* Chiruzzo et al., 2022), we experimented with injecting PoS features to the SL input for this translation direction. As LSE does not yet have a language model for tokenisation, tagging, and parsing, we used (SpaCy) tags from Spanish

where a gloss is labelled with the same lexical item from Spanish aligned using the fast_align model (Dyer et al., 2013). The results from these experiments were generally negligible or negative. We attribute this to the fact that, other than sharing lexemes, the underlying structure of Spanish and (glossed) LSE are typologically very different. Some LSE glosses such as "IGUAL" (*like*, adverb) share neither meaning nor grammatical category with their homograph "igual" (*equal*, adjective) in Spanish. The grammatical structure of each language also differs greatly such as word ordering based on constituents and semantic roles.

In this respect, we conducted further experiments where we manually PoS-tagged LSE glosses from the iSignos corpus based on their entries in a gloss lexicon[7]. We included these tags as features in G2T translation models and, this time, gained improvements (BLEU-4 metric) across all experimental settings when including glosses versus glosses alone as input (McGill et al., 2023). We also attempted to train a zero-shot PoS-tagger[8] with iSignos manually tagged data, but this yielded suboptimal results as it only output tags of VERB, PRON, and NOUN for all glosses.

We aim to refine this approach with our future research direction into creating pre-trained word embedding representations for LSE (McGill, *forthcoming*) and use these while training tagging and translation models for LSE. Also, we continue collaboration with the University of Vigo in their steps towards constructing a UD Treebank model for LSE (García-Miguel and Cabeza, 2019). This type of work, we predict, will bring NLU capabilities for SLs one step closer and therefore allow us to not only include spoken languages in the NLU module as has been the case for SignON.

4. Word and Sign Sense Disambiguation

Word Sense Disambiguation is a very relevant task inside NLU, and also in the case of SLs. A word can refer to many very different meanings and, in SLs, each of these meanings can be expressed by very different signs. For example, in spoken Spanish the word "banco" could mean "bench" or "bank" depending on the context; meanwhile in LSE these two concepts are expressed with two different signs as Figure 1 shows. Therefore, disambiguating the source message will considerably impact the production of the sign translation output.

[7] http://isignos.uvigo.es/es/lexico
[8] https://github.com/jiesutd/NCRFpp

Figure 1. Dilse[9] LSE sign production for the signs “bench” (left) and “bank” (right)

Therefore, we have included a WSD in our pipeline after analysing different solutions. The different tools considered are:

- Amuse-WSD (Orlando et al., 2021) is a tool which integrates Large Language Models trained for multilingual Word Sense Disambiguation. It can be used through an API and also is distributed as a dockerised package, which hinders the integration in our own pipeline. Furthermore, integrating these models could increase the delays in the translation process.
- LESK algorithm follows a classic approach that heuristically searches for the most likely meaning for the word (Lesk, 1986). Although a long-established algorithm, its performances are relatively limited, reporting an accuracy of 50%-70% (Lesk, 1986).
- WordNet Path Similarity uses the connections between syntactic sets in WordNet to match the correct word sense and the sentence context according to the shortest path[10]. This solution is promising due to its availability for most of the languages covered and its good performances.

We compared the LESK and WordNet Path Similarity methods for several example sentences extracted from the HoReCo dataset. Amuse-WSD is discarded because it runs in an external service, which is not convenient for a fully-local implementation. After a comparison between approaches, we found the WordNet Path Similarity to perform better than LESK. Namely, we used WordNet 2022 for this software and it can be easily upgraded to new WordNet versions. The WSD method is only applied to certain PoS elements, those which can be ambiguous such as verbs, nouns and adjectives. Table 4 presents some results for the solution adopted.

[9] https://fundacioncnse-dilse.org
[10] https://www.nltk.org/_modules/nltk/corpus/reader/wordnet.html#Synset.path_similarity

Table 4. TextNormalizer outputs. "R" denotes the original sentence from HoReCo and "S" the names of the synsets from WordNet 2022

ENGLISH **R:** Awesome! I will come back! **W:** amazing.s.02|PUNCT|PRON|AUX|issue_forth.v.01|ADV|PUNCT **R:** In town to Christmas shop and spend time with family. **S:** ADP|town.n.02|ADP|christmas.n.01|shop.n.01|CCONJ|stand.v.03|time.n.05|ADP|class.n.01|PUNCT
SPANISH **R:** ¡Increíble! ¡Volveré! **W:** PUNCT|incredible.a.01|PUNCT|PUNCT|turn.v.10|PUNCT **R:** En la ciudad para hacer compras navideñas y estar en familia. **S:** ADP|DET|township.n.01|ADP|produce.v.02|purchase.n.02|ADJ|CCONJ|be.v.03|ADP|kin.n.02|PUNCT
DUTCH **R:** Geweldig! Ik kom zeker terug! **S:** ADJ|PUNCT|PRON|appear.v.02|ADJ|ADV|PUNCT **R:** We waren in de stad om kerstinkopen te doen en tijd door te brengen met familie **S:** PRON|AUX|ADP|DET|NOUN|ADP|NOUN|ADP|work.v.01|CCONJ|time.n.05|ADP|ADP|while_away.v.01|ADP|family.n.06

Discussion & Limitations. From Table 4, we can observe that the WSD module is able to deal with the most ambiguous words for Spanish and English, while the error rate for Dutch is higher. This fact is much more notable in the results presented in Annex 1. Some of these failures have two main reasons:

(1) TextNormalizer sometimes introduces errors that propagate through the pipeline and causes the WSD to not match the correct, exact meaning. A clear example of this fact is the word "spend" in the second English sentence, it was replaced by "stand" (see Table 2) leading to an incorrect sense matching.

(2) The employed method is not able to catch the whole context of the word in the sentence. For instance, the Spanish "ciudad", which means "town" is associated with the meaning "township" for the second Spanish sentence.

Regarding Irish, we did not find any resource and/or tool for this language and we had to exclude this language from the WSD module. In Annex 1, more examples are shown and the results are discussed in more detail.

5. Architecture of the NLP Pipeline

The combination of the processes described above is implemented in the class SignON_NLP available in a GitHub repository [11]. This class manages the communication with the pipeline through an API server using the FLASK[12] library and executes the different processes on the source text. The implementation scheme is shown in Figure 2.

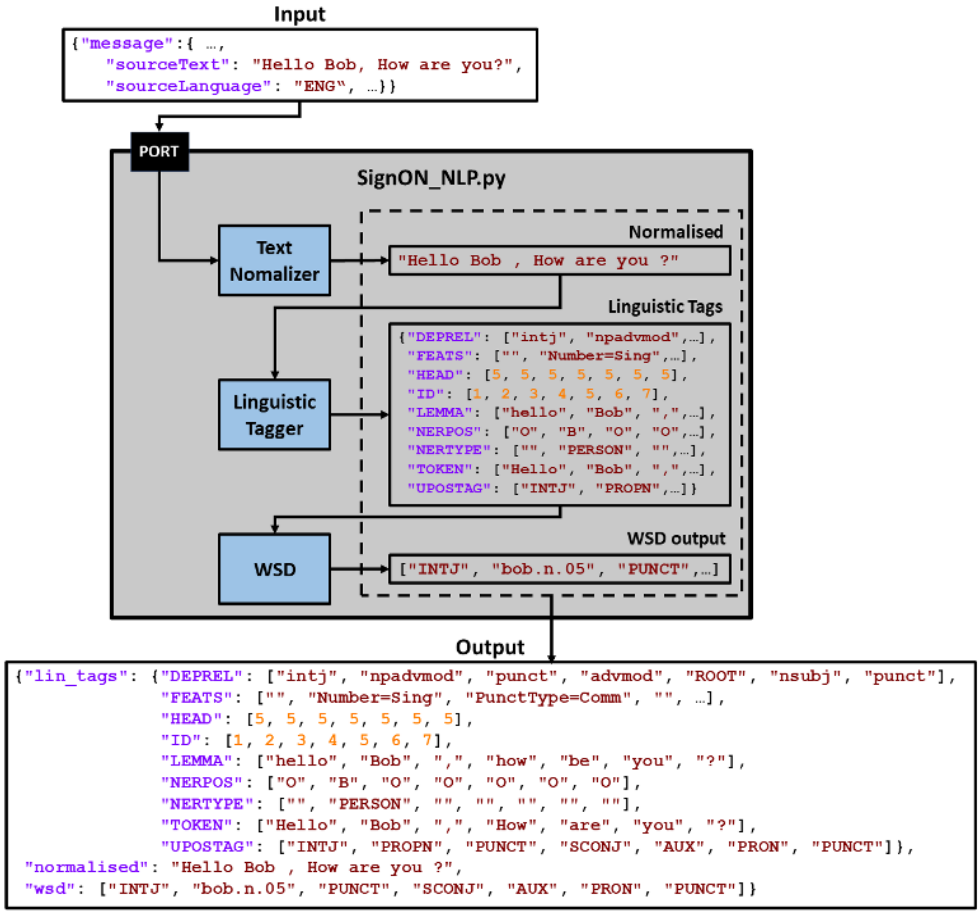

Figure 2. Block diagram of the NLP pipeline

The system receives the source data in the form of JSON from previous modules through an open port. The source language and text are unencapsulated and fed to the TextNormalizer. The linguistic features are computed on the normalised text and they serve as input to the WSD module. Namely, the WSD module uses the values in the “TOKEN” and “UPOSTAG” fields to assign the WordNet synsets. All module outputs are compiled in an output dictionary that is served as response.

[11] https://github.com/signon-project-wp3/WP3-Second-NLP-Pipeline
[12] https://flask.palletsprojects.com/en/3.0.x/

6. Conclusion

This document presents the second NLP pipeline for the SignON project. The previous version presented in D3.5 "First Natural Language Processing pipeline has been substantially upgraded to enable text normalisation and WSD. We were able to provide support to English, Spanish, Irish and Dutch for text normalisation and English, Spanish and Dutch for WSD. Text normalisation is implemented to remove non-informative characters and punctuations, and to correct some typos that the user could accidentally produce while writing. Regarding WSD, the objective is to annotate words that could have ambiguous meanings with the more accurate WordNet meaning. Additionally, the pipeline includes linguistic tagging already reported in the previous pipeline version.

We performed a test on HoReCo samples which brought into light some limitations in different processes presented. We found that the spell checker module sometimes produces errorful outputs due its simplicity. The method deployed only checks words in a dictionary without taking into account the sentence context or word category. This process could be enhanced by introducing some context in the search, however we have not found solutions suitable for our pipeline. In the case of WSD, we found positive performances (see Annex 1), but some meanings are matched to close ones and not the exact one. Generally, the performances of text normalisation and WSD drop for Dutch and Irish, and for the latter we were not able to implement WSD.

In terms of the requirements laid out in the Grant Agreement for T3.5, the NLU pipeline fulfils the brief in terms of its place in the SignON ecosystem. We opted for statistical and less computationally-demanding methods for linguistic tagging and parsing such as by using the 'medium' or 'small' sized models provided by SpaCy[13], as well as for text normalisation and word sense disambiguation. It was, however, not possible to provide figurative language handling or coreference resolution as these models are either incompatible with our pipeline architecture, only available for English, and/or only available in computationally-demanding large neural models.

For future iterations of an NLU module in similar projects, it may be preferable to permit more computationally-expensive approaches as these tend to be better-performing, include extended functionality, and use more up-to-date methods. For example, and related to the functionality of the

[13] https://spacy.io/models

SignON NLU pipeline, it would be possible to use multilingual, transformer-based models like SensEmBERT (Scarlini et al., 2020) for word sense disambiguation. In addition, there has been exciting developments in recent years towards the tagging and parsing of SL (gloss) data such as the development of Universal Dependencies[14] treebanks for Swedish Sign Language (Östling et al., 2017), Italian Sign Language (Caligiore, 2020) and Turkish Sign Language (Eryiğit et al., 2020). There is also work underway for a UD treebank for LSE (García-Miguel. and Cabeza, 2019). This would allow the development of a similar pipeline that we have created for SignON which can perform similar tasks for SL glosses.

[14] https://universaldependencies.org/swl/index.html

References

Armegnol Estapé, J. and Costa-jussà, M. R. (2021) Semantic and syntactic information for neural machine translation: Injecting features to the transformer. *Machine Translation* 33:3 *3*, pp. 3-17

Caligiore, G. (2020) *Universal Dependencies for Italian Sign Language: a treebank from the storytelling domain* [Doctoral dissertation, Università degli Studi di Torino]

Chiruzzo, L., McGill, E., Gómez, S. E., & Saggion, H. (2022) Translating spanish into spanish sign language: Combining rules and data-driven approaches. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages* (LoResMT 2022), pp. 75-83.

De Sisto, M., Vandeghinste, V., Egea Gómez, S., De Coster, M., Shterionov, D., & Saggion, H. (2022) Challenges with sign language datasets for sign language recognition and translation. In *13th International Conference on Language Resources and Evaluation* (LREC 2022)

Dyer, C., Chahuneau, V., and Smith, N. A. (2013) A simple, fast, and effective reparameterization of IBM model 2. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 644-648

Egea-Gómez S. et al. (2021) Syntax-aware Transformers for Neural Machine Translation: The Case of Text to Sign Gloss Translation. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora* (BUCC 2021), pp. 18–27

Egea-Gómez S. et al. (2022) Linguistically Enhanced Text to Sign Gloss Machine Translation. In: Natural Language Processing and Information Systems. NLDB 2022. Lecture Notes in Computer Science, vol 13286. Springer, Cham.

Eryiğit, G., Eryiğit, C., Karabüklü, S, Kelepir, M., Özkul, A., Pamay, T., Torunoğlu-Selamet, D., and Köse, H. (2020) Building the first comprehensive machine-readable Turkish sign language resource: methods, challenges and solutions. *Language Resources and Evaluation* 54, 97-121.

García-Miguel, J. M. and Cabeza, C. (2019) Hacia un TreeBank de dependencias para la LSE. *Hesperia. Anuario de filología hispánica XXII-2,* pp. 111-143

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th annual international conference on Systems documentation (SIGDOC '86), pp.24–26. ACM, New York, USA.

McGill, E., Chiruzzo, L., Egea-Gómez, S., and Saggion, H. (2023) Part-of-Speech tagging Spanish Sign Language data and its applications in Sign Language machine translation. In: *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023)*, pp. 70-76

McGill, E. (*forthcoming*) *Bootstrapped pre-trained embeddings for Spanish Sign Language glosses*

Moryossef, A., Yin, K., Neubig, G., and Goldberg, Y. (2021) Data Augmentation for Sign Language Gloss Translation https://arxiv.org/abs/2105.07476

Orlando, R., Conia, S., Brignone F., Cecconi, F., and Navigli, R. (2021). AMuSE-WSD: An All-in-one Multilingual System for Easy Word Sense Disambiguation. In Proceedings of the 2021 Conference on EMNLP, pp. 298–307. Online and Punta Cana, Dominican Republic. ACL.

Östling, R., Börstell, C., Gärdenfors, M., and Wirén, M. (2017) Universal Dependencies for Swedish Sign Language. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 303-308. Gothenburg, Sweden.

Scarlini, B., Pasini, T., and Navigli, R. (2020) Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In: *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(05), 8758-8765. https://doi.org/10.1609/aaai.v34i05.6402

Sennrich, R. and Haddow, B. (2016) Linguistic Input Features Improve Neural Machine Translation. In: *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pp.83-91, Berlin, Germany.

Sutton-Spence, R., & Woll, B. (1999). *The linguistics of British Sign Language: an introduction. Cambridge University Press.*

Annex 1. Testing the NLU on HoReCo samples

In order to assess the performance of the processes inside the NLP pipeline, we conducted a test on the 15 sample postings of the corpus HoReCo[15]. This data used was generated in the scope of this project suiting the project domain. As we had to manually check the outputs generated, we limit the test to teen samples.

The test framework is performed as follows:

(1) The original sentences from HoReCo are manually altered to introduce typos and not allowed punctuations.
(2) The altered version of the sentences are inputted to the TextNormalizer.
(3) The normalised text is introduced to the LinguisticTagger.
(4) And finally, the linguistics information is used to annotate the WordNet synsets in the WSD module.

Tables A1-4 present the outputs from all the NLP modules for English, Spanish, Dutch and Irish, respectively. In all these tables, “R” denotes the original HoReCo sentence, “I” the sentence inputted to the TextNormalizer, “N” the normalised text and “S” the synset associated with each word. Note that we were not able to implement WSD for Irish. For the sake of result interpretability, the meaning of the WordNet synsets generated in the test are attached at the end of this annex.

From Table A-1, we observed that the TextNormalizer is able to correct most typos and alterations for English. Namely, we introduced 15 alterations and the normalised text only contains 4 errors. While on WSD, only 6 (out of 36) ambiguous concepts were wrongly disambiguated. From Table A-2, we see that more than half of alterations were corrected (7 out of 13); and only 13 (from 39) terms were wrongly disambiguated. In the case of Dutch (Table A3), the success rate in the normalisation increases, since we introduced 13 alterations in the text and the normalised texts only contain 2 errors. On the contrary, WSD for Dutch is much improvable, we were able to correctly disambiguate 15 terms from a total of 38. The worst results are found for Irish (Table A4), only 6 errors were corrected by the normaliser from a total of 18. For this language, the WSD is not implemented.

[15] https://european-language-equality.eu/wp-content/uploads/2023/04/ELE2_Project_Report_NGT_HoReCo.pdf

Table A1. Outputs for HoReCo English samples.

I: Enjoiable.
N: Enjoyable .
R: Enjoyable
S: enjoyable.s.01|PUNCT

I: Cockroaches everywhere !!!!!~@!
N: Cockroaches everywhere !
R: Cockroaches everywhere!
S: cockroach.n.01|ADV|PUNCT

I: Awesume!!!!!! I will come back !
N: Awesome ! I will come back !
R: Awesome! I will come back!
S: amazing.s.02|PUNCT|PRON|AUX|issue_forth.v.01|ADV|PUNCT

I: Excellent room and welll mannered service!
N: Excellent room and well mannered service !
R: Excellent room and well mannered service!
S: excellent.s.01|room.n.01|CCONJ|ADV|mannered.s.01|service.n.11|PUNCT

I: The evening help was rude and none atentive.
N: The evening help was rude and none tentative .
R: The evening help was rude and none attentive.
S: DET|evening.n.03|aid.n.02|AUX|uncivil.a.01|CCONJ|none.n.02|probationary.s.01|PUNCT

I: The hotel was perfect for our girtfriend get-away.
N: The hotel was perfect for our girlfriend getaway .
R: The hotel was perfect for our girlfriend get-away.
S: DET|hotel.n.01|AUX|perfect.s.03|ADP|PRON|girlfriend.n.02|pickup.n.05|PUNCT

I: In town to Christmass shop and spand time with family.
N: In town to Christmas shop and stand time with family .
R: In town to Christmas shop and spend time with family.
S: ADP|town.n.02|ADP|christmas.n.01|shop.n.01|CCONJ|stand.v.03|time.n.05|ADP|class.n.01|PUNCT

I: Very comfortable roms, will stay again when in the area.
N: Very comfortable rooms , will stay again when in the area .
R: Very comfortable rooms, will stay again when in the area.
S: ADV|comfortable.s.05|room.n.01|PUNCT|AUX|stay.v.05|ADV|SCONJ|ADP|DET|area.n.05|PUNCT

I: Very restful and quite.... Also had a nice pool to relax in. $
N: Very restful and quite . Also had a nice pool to relax in .
R: Very restful and quite. Also had a nice pool to relax in.
S: ADV|restful.a.01|CCONJ|ADV|PUNCT|ADV|own.v.01|DET|nice.s.03|pool.n.06|PART|relax.v.07|ADP|PUNCT

I: The hutel was beautiful and the stafff was awesome. One of the best beeches in mexico
N: The hotel was beautiful and the staff was awesome . One of the best beeches in Mexico
R: The hotel was beautiful and the staff was awesome. One of the best beaches in mexico
S: DET|hotel.n.01|AUX|ADJ|CCONJ|DET|staff.n.04|AUX|amazing.s.02|PUNCT|NUM|ADP|DET|good.s.21|beech.n.02|ADP|mexico.n.01

Table A2. Outputs for HoReCo Spanish samples.

R: Adorable.
I: Adorable. =@
N: Adorable .
S: lovable.a.01|PUNCT

R: ¡Hay cucarachas por todas partes!
I: ¡Hay cucarachas por todes partes!
N: ¡ Hay cucarachas por todos partes !
S: PUNCT|AUX|NOUN|ADP|DET|part.n.09|PUNCT

R: ¡Increíble! ¡Volveré!
I: ¡Increíble! ¡Bolveré!
N: ¡ Increíble ! ¡ Volveré !
S: PUNCT|incredible.a.01|PUNCT|PUNCT|turn.v.10|PUNCT

R: Excelente habitación y muy buen servicio.
I: Excelente habitación y muy buen servicio....
N: Excelente habitación y muy buen servicio .
S: top-flight.s.01|room.n.01|CCONJ|ADV|pretty.s.02|facility.n.01|PUNCT

R: Los trabajadores de la tarde fueron maleducados y poco atentos.
I: Los trabajadores de la tarde fueron maleducados y poco attentos.
N: Los trabajadores de la tarde fueron maleducados y poco atento .
S: DET|worker.n.01|ADP|DET|afternoon.n.01|AUX|impolite.a.01|CCONJ|ADV|mindful.a.01|PUNCT

R: El hotel fue perfecto/ideal para nuestra escapada en pareja.
I: El hutel fue ideal para nuestra escapada en pareja.
N: El hotel fue ideal para nuestra escapada en pareja .
S: DET|hotel.n.01|AUX|ideal.a.03|ADP|DET|escapade.n.02|ADP|mate.n.03|PUNCT

R: En la ciudad para hacer compras navideñas y estar en familia.
I: En la ciudad para hacer compras navidenas y estar en familia.
N: En la ciudad para hacer compras navideñas y estar en familia .
S: ADP|DET|township.n.01|ADP|produce.v.02|purchase.n.02|ADJ|CCONJ|be.v.03|ADP|kin.n.02|PUNCT

R: Las habitaciones son muy cómodas, volveré cuando esté otra vez por la zona.
I: Las habitaciones son muy comodas, volvere cuando este otra vez por la zona.
N: Las habitaciones son muy como-das , volver cuando este otra vez por la zona .
S: DET|room.n.01|AUX|ADV|ADJ|PUNCT|turn.v.10|SCONJ|DET|DET|time.n.01|ADP|DET|area.n.05|PUNCT

R: Lugar apacible y tranquilo. También dispone de una piscina para desconectar.
I: Lugar apacible y trankilo.... Tambien dispone de una piscina para desconectar.
N: Lugar apacible y tranquilo . También dispone de una piscina para desconectar .
S: location.n.01|placid.s.01|CCONJ|unexcited.a.01|PUNCT|ADV|fix.v.12|ADP|DET|swimming_pool.n.01|ADP|disconnect.v.02|PUNCT

R: El hotel era precioso y los trabajadores increíbles. Una de las mejores playas de México.
I: El hotel era precioso y los trabajadores increíbles. Una de las mejores platjas de México.
N: El hotel era precioso y los trabajadores increíbles . Una de las mejores platas de México .
S: DET|hotel.n.01|AUX|beautiful.a.01|CCONJ|DET|worker.n.01|incredible.a.01|PUNCT|PRON|ADP|DET|better.a.02|silver.n.01|ADP|PROPN|PUNCT

Table A3. Outputs for HoReCo Dutch samples.

R: Heb genoten.
I: Heb genotenn.
N: Heb genoten .
S: AUX|enjoy.v.01|PUNCT

R: Overal kakkerlakken!
I: Overal kakkerlakken!!!!!!!&/(
N: Overal kakkerlakken !
S: ADV|NOUN|PUNCT

R: Geweldig! Ik kom zeker terug!
I: Geweldig!!!!!! Ik kom zekerr terug!
N: Geweldig ! Ik kom zeker terug !
S: ADJ|PUNCT|PRON|appear.v.02|ADJ|ADV|PUNCT

R: Uitstekende kamer en goede service!
I: Uitstekende kamer en goede service!
N: Uitstekende kamer en goede service !
S: PROPN|room.n.01|CCONJ|possession.n.02|service.n.13|PUNCT

R: De avondmedewerker was onbeschoft en niet attent.
I: De avondmedewerkar wes onbeschoft en niet atttent.
N: De avondmedewerker Wes onbeschoft en niet attent .
S: DET|NOUN|PROPN|PROPN|CCONJ|ADV|ADJ|PUNCT

R: Het hotel was perfect voor ons vriendinnen uitje.
I: Het hotel was perfect voor ons vriendinnen uitje.
N: Het hotel was perfect voor ons vriendinnen uitje .
S: DET|hotel.n.01|AUX|ADJ|ADP|PRON|lover.n.01|NOUN|PUNCT

R: We waren in de stad om kerstinkopen te doen en tijd door te brengen met familie.
I: We waren in de stud om kerstinkopen te doen en tijd door te brengen met familia
N: We waren in de stud om kerstinkopen te doen en tijd door te brengen met familie
S: PRON|AUX|ADP|DET|NOUN|ADP|NOUN|ADP|work.v.01|CCONJ|time.n.05|ADP|ADP|while_away.v.01|ADP|family.n.06

R: Zeer comfortabele kamers, ik zal er zeker opnieuw verblijven wanneer ik weer in de buurt ben.
I: Zear comfortabele kamers, ik zal er zeker opnieuw verbliiven wanneer ik weer in de buurt ben.
N: Zeer comfortabele kamers , ik zal er zeker opnieuw verblijven wanneer ik weer in de buurt ben .
S: ADV|ADJ|room.n.04|PUNCT|PRON|AUX|ADV|ADJ|ADV|bide.v.01|SCONJ|PRON|ADV|ADP|DET|village.n.02|be.v.01|PUNCT

R: Heel rustgevend en rustig. Er was ook een mooi zwembad om in te ontspannen.
I: Heel rustgevend en rustig..... Er was ookj een mooi zwembad om in te ontspannen.
N: Heel rustgevend en rustig . Er was ook een mooi zwembad om in te ontspannen .
S: ADJ|VERB|CCONJ|ADJ|PUNCT|ADV|exist.v.01|ADV|DET|ADJ|swimming_pool.n.01|ADP|ADP|ADP|relax.v.04|PUNCT

R: Het hotel was prachtig en het personeel was geweldig. Een van de beste stranden in Mexico.
I: Het hotel was prchtig en het personeil was geweldig. Een van de beste stranden in Mexico.
N: Het hotel was prachtig en het personeel was geweldig . Een van de beste stranden in Mexico .
S: DET|hotel.n.01|AUX|ADJ|CCONJ|DET|work_force.n.01|AUX|ADJ|PUNCT|NUM|ADP|DET|possession.n.02|beach.n.01|ADP|mexico.n.01|PUNCT

Table A4. Outputs for HoReCo Irish samples.

R: Taitneamhach
I: Taitneamhach**h**
N: **Neamhthaitneamh**

R: Ciaróga dhubha i ngach áit!
I: Ciaró**j**a dhubha i ngach áit !**!!!!**
N: Ciaró**g**a dhubha i ngach áit **!**

R: Iontach ! Tiocfaidh mé ar ais!
I: Iontach!**!!!!!** Ti**u**cfaidh mé ar ais!
N: Iontach ! **Faitidh** mé ar ais !

R: Seomra iontach, agus seirbhís dea-bhéasach.
I: Seomra **o**ntach, agus seirbhís dea-**h**éasach. **$)&**
N: Seomra **a**ontach**as** , agus seirbhís **deasach** .

R: Bhí an fhoireann tráthnóna drochbhéasach, agus níor thug siad aird dúinn.
I: Bhí an**m** fhoireannn tráthnóna drochbhéasach, agus níor thog siad aird dúinn.
N: Bhí an **f**oirean**n** tráthnóna drochbhéasach , agus níor thor siad aird dúinn .

R: Bhí an t-óstán foirfe dár laethanta saoire chailíní.
I: **Bí** an **tó**stán fo**j**rfe dár laethanta saoire chailíní.
N: **Bí** an **t**-óstán fo**i**rfe dár laethanta saoire chailíní .

R: Tá mé ar an mbaile chun siopadóireacht a dhéanamh don Nollaig, agus am a chaitheamh leis an teaghlach.
I: T**a** m**e** ar an mbaile chun siopadóireacht a dhéanamh don Nollaig, agus am a chaitheamh leis an teaghlach.
N: **Na de** ar an mbaile chun siopadóireacht a dhéanamh don Nollaig , agus am a chaitheamh leis an teaghlach .

R: Seomraí an-chompordach. D'fhanfainn arís dá mbeinn sa cheantar.
I: Seomraí an-chompordach**.....** D'fhanfainn arís da mbeinn sa cheantar.
N: Seomraí **míchompordaí** . D'fhanainn arís a mbeinn sa cheantar .

R: An-shuaimhneach agus ciúin. Bhí linn snámha deas ann freisin chun scíth a ligean ann.
I: An-shuaimhneach agus ciuin. Bhí linn snámha de**e**s ann freisin chun scíth a ligean ann.
N: **Suaimhneacha** agus cluin . Bhí linn snámha **Des** ann freisin chun scíth a ligean ann .

R: Bhí an t-óstán go hálainn, agus bhí an fhoireann ar fheabhas. Tá sé ar cheann de na tránna is fearr i Meicsiceo
I: Bhí an **tó**stán go hálainn, agus bhí an fhoireann ar fheabhas. Tá sé ar c**hh**eann de na tránna is fear i Meicsiceo
N: Bhí an **t**-óstán go hálainn , agus bhí an fhoireann ar fheabhas . Tá sé ar **ceannach** de na tránna is fear i Meicsiceo

WordNet 2022 meanings:

afternoon.n.01: the part of the day between noon and evening

aid.n.02: the activity of contributing to the fulfillment of a need or furtherance of an effort or purpose

amazing.s.02: inspiring awe or admiration or wonder; ; ; ; - Melville

appear.v.02: come into sight or view

area.n.05: a part of a structure having some specific characteristic or function

be.v.01: have the quality of being; (copula, used with an adjective or a predicate noun)

be.v.03: occupy a certain position or area; be somewhere

beach.n.01: an area of sand sloping down to the water of a sea or lake

beautiful.a.01: delighting the senses or exciting intellectual or emotional admiration

beautiful.s.02: (of weather) highly enjoyable

beech.n.02: wood of any of various beech trees; used for flooring and containers and plywood and tool handles

better.a.02: (comparative of `good') changed for the better in health or fitness

bide.v.01: dwell

christmas.n.01: period extending from Dec. 24 to Jan. 6

class.n.01: a collection of things sharing a common attribute

cockroach.n.01: any of numerous chiefly nocturnal insects; some are domestic pests

comfortable.s.05: in fortunate circumstances financially; moderately rich

disconnect.v.02: make disconnected, disjoin or unfasten

enjoy.v.01: derive or receive pleasure from; get enjoyment from; take pleasure in

enjoyable.s.01: affording satisfaction or pleasure

escapade.n.02: any carefree episode

evening.n.03: the early part of night (from dinner until bedtime) spent in a special way

excellent.s.01: very good; of the highest quality

exist.v.01: have an existence, be extant

facility.n.01: a building or place that provides a particular service or is used for a particular industry

family.n.06: (biology) a taxonomic group containing one or more genera

fix.v.12: make ready or suitable or equip in advance for a particular purpose or for some use, event, etc

girlfriend.n.02: a girl or young woman with whom a man is romantically involved

good.s.21: generally admired

hotel.n.01: a building where travelers can pay for lodging and meals and other services

ideal.a.03: of or relating to the philosophical doctrine of the reality of ideas

impolite.a.01: not polite

incredible.a.01: beyond belief or understanding

issue_forth.v.01: come forth

kin.n.02: group of people related by blood or marriage

location.n.01: a point or extent in space; a point or extent in space

lovable.a.01: having characteristics that attract love or affection

lover.n.01: a person who loves someone or is loved by someone

mannered.s.01: having unnatural mannerisms

mate.n.03: the partner of an animal (especially a sexual partner)

mexico.n.01: a republic in southern North America; became independent from Spain in 1810

mindful.a.01: bearing in mind; attentive to

nice.s.03: done with delicacy and skill

none.n.02: a service in the Roman Catholic Church formerly read or chanted at 3 PM (the ninth hour counting from sunrise) but now somewhat earlier

own.v.01: have ownership or possession of

part.n.09: one of the portions into which something is regarded as divided and which together constitute a whole

perfect.s.03: precisely accurate or exact

pickup.n.05: the attribute of being capable of rapid acceleration

placid.s.01: (of a body of water) free from disturbance by heavy waves

pool.n.06: a small body of standing water (rainwater) or other liquid

possession.n.02: anything owned or possessed

pretty.s.02: (used ironically) unexpectedly bad

probationary.s.01: under terms not final or fully worked out or agreed upon

produce.v.02: create or manufacture a man-made product

purchase.n.02: something acquired by purchase

relax.v.04: cause to feel relaxed

relax.v.07: become less severe or strict

restful.a.01: affording physical or mental rest

room.n.01: an area within a building enclosed by walls and floor and ceiling

room.n.04: the people who are present in a room

service.n.11: (law) the acts performed by an English feudal tenant for the benefit of his lord which formed the consideration for the property granted to him

service.n.13: the act of delivering a writ or summons upon someone

shop.n.01: a mercantile establishment for the retail sale of goods or services

silver.n.01: a soft white precious univalent metallic element having the highest electrical and thermal conductivity of any metal; occurs in argentite and in free form; used in coins and jewelry and tableware and photography

staff.n.04: building material consisting of plaster and hair; used to cover external surfaces of temporary structure (as at an exposition) or for decoration

stand.v.03: occupy a place or location, also metaphorically

stay.v.05: remain behind

swimming_pool.n.01: pool that provides a facility for swimming

time.n.01: an instance or single occasion for some event

time.n.05: the continuum of experience in which events pass from the future through the present to the past

top-flight.s.01: excellent; best possible

town.n.02: the people living in a municipality smaller than a city

township.n.01: an administrative division of a county

turn.v.10: cause to move around a center so as to show another side of

uncivil.a.01: lacking civility or good manners; - Willa Cather

unexcited.a.01: not excited

village.n.02: a settlement smaller than a town

while_away.v.01: spend or pass, as with boredom or in a pleasant manner; of time

work.v.01: exert oneself by doing mental or physical work for a purpose or out of necessity; work

work_force.n.01: the force of workers available

worker.n.01: a person who works at a specific occupation

Annex 2. Regex rules used in the text normaliser

Table A.5 presents the Regex rules used in text normalisation.

Table A5. Regex rules in the TextNormalizer.

```
ALLOWED_PUNCT = r'.,!?¿¡'
NOT_ALLOWED_PUNCT = r'"#$%&\'()*+-/:;<=>@[\\]^_`{|}~'
# REMOVES NOT ALLOWED PUNCTUATION
sentence = ''.join([c if c not in NOT_ALLOWED_PUNCT else '' for c in sentence])
# REMOVES EXTRA PUNCTUATIONS: 'Ahh!!!' => 'Ahh!'
sentence = re.sub(r'(\W)(?=\1)', '', sentence)
# ADDES WHITESPACE BEFORE/AFTER PUNCTUATIONS: 'Ahh!' => 'Ahh !'
sentence = re.sub(r'(['+ALLOWED_PUNCT+'])', r' \1 ', sentence)
# REMOVES MULTIPLE WHITESPACES
sentence = re.sub(r' +', r' ', sentence)
# REMOVES WHITESPACE AT THE END OF A SENTENCE
sentence = sentence if sentence[-1] != ' ' else sentence[:-1]
# REMOVES WHITESPACE AT THE BEGINNING OF A SENTENCE
sentence = sentence if sentence[0] != ' ' else sentence[1:]
```