Research and Innovation Action (RIA) H2020 – 957017

Stream Learning for Multilingual Knowledge Transfer

https://selma-project.eu

D3.6 Intermediate Release of Post-editing and User Feedback Capabilities

Work Package	3
Responsible Partner	LIA
Author	Yannick Estève
Contributors	Tugtekin Turan
Reviewer	Afonso Mendes
Version	1.0
Contractual Date	31 December 2022
Delivery Date	22 December 2022
Dissemination Level	Public

Version History

Version	Date	Description
0.1	15/11/2022	Initial Table of Contents (ToC)
0.2	19/12/2022	Input
0.3	21/12/2022	Internal Review
1.0	22/12/2022	Finalization and Submission

Executive Summary

This deliverable describes the intermediate release of software components developed within WP3, particularly regarding post-editing and user feedback capabilities. The final release at the end of the project will follow this document.

SELMA's approach to speech and language processing is targeting both low and high resourced languages

Table of Contents

Table of Figures

1. Introduction

In this report, we detail the work made by SELMA to take into account post-editing and user feedback. The software is not mature enough to be released at this stage, and the central part of the work is still being made at the research level.

We present in this document our first software that allows us to inject linguistic information from text into an end-to-end neural ASR model. To our knowledge, this is the first software that offers this possibility for such technology. We are still working on this approach, as well as on beam search decoding driven by language models, that allows us to inject user feedback as textual entries.

We also investigated state-of-the-art to consider the integration of user feedback, and we expect now to explore the use of a translation memory-like approach.

2. In Progress Software

2.1 Toward the Injection of User Feedback and Linguistic Information

Named entity recognition from speech consists in recognizing words from speech, detecting word sequences that support a named entity, and categorizing this entity. End-to-end neural approaches suffer from the lack of paired audio and textual data with a named entity annotation. An end-to-end model for named entity recognition from speech without paired training data has been built in the framework of the SELMA project. This success opens new perspectives to update the linguistic information into a pre-trained end-to-end ASR model. Such linguistic information could be obtained from user feedback or daily textual news.

Our approach is based on the use of an external model (named Text-to-ASR-Embeddings model) to generate a sequence of vectorial representations from text, similar to the ASR hidden representations. A NER-S module is then trained using these representations as input and the annotated existent textual data as output.

This ASR model is based on the software material shared here: https://github.com/SELMA-project/LIA_speech/tree/main/asr.

The Text-to-ASR-embeddings model is used to mimic the 80-dimensional embeddings of ASR. It is based on the Tacotron2 neural architecture for speech synthesis (text-to-speech). The NER module consists of a BiLSTM model composed of 5 BiLSTM layers, with 512 dimensions each. Experiments on the QUAERO corpus show that this approach is very promising. Our paper has been submitted, accepted, and published at INTERSPEECH 2022[1].

[1] https://interspeech2022.org

2.2 Injecting Textual Data into end-to-end speech-to-text Models

The contribution presented in the previous section is a step toward the first direction planned in the scientific documentation of the SELMA project. Thanks to this success, we are investigating novel ways to generate massive amounts of training data for the post-editing task. Indeed, the nature of speech processing models becomes increasingly *end-to-end*, *i.e.*, the downstream task (machine translation, transcription, named entity recognition, *etc.*) is processed directly from the speech by a single large neural model. Thus, it is hard to inject textual data into such models in order to update their knowledge, while user feedback is mainly based on text. We expect to adapt the approach presented in section 2.1 in order to inject such textual user feedback into a training process that updates only the weights of a high layer of the end-to-end neural network.

Another processed approach consists of applying an adaptable language model during a beam search decoding on the top of the end-to-end neural network to drive the decoding process by textual data. LIA is a consequent contributor to the SpeechBrain toolkit, and this beam search decoding driven by language models is being implemented: see https://github.com/speechbrain/speechbrain/pull/751#issuecomment-1330308097 for more details. Using a language model that could be updated directly from the text makes it possible to constrain the vocabulary and add words or remove wrong spellings.

2.3 Evaluating the Information from User-corrected Data in Speaker Tasks

The traditional approach to updating a speaker recognition system with user feedback is to acquire a large amount of training data for the new speaker, combine it with the data used to train the original classifier, and then retrain the classifier on the combined data. However, this approach requires a significant amount of data and can be time-consuming. To add a new user to a speaker recognition system with user feedback in a more efficient way, we propose the following steps:

1. Collecting feedback: The first step is to collect user feedback on the system's performance through a user interface, such as a web application. This can include suggestions for speaker modifications or the addition of new speakers.
2. Adding new data: Using the collected user feedback, we can identify the remaining speaker segments in the recordings and update them with the new or modified speaker data. This helps to expand the scope of the limited user feedback and provide the classification system with up-to-date data.
3. Updating model: After adding the new user data, we can use techniques such as fine-tuning or incremental learning to update the speaker model and improve its performance. We propose using the latter approach in order to avoid retraining the model on all the data again.

As a first step, we have developed a webservice to collect user data, as it shown in the following figure, where the user can change a speaker's label or add a new speaker to the system over a diarized transcription.

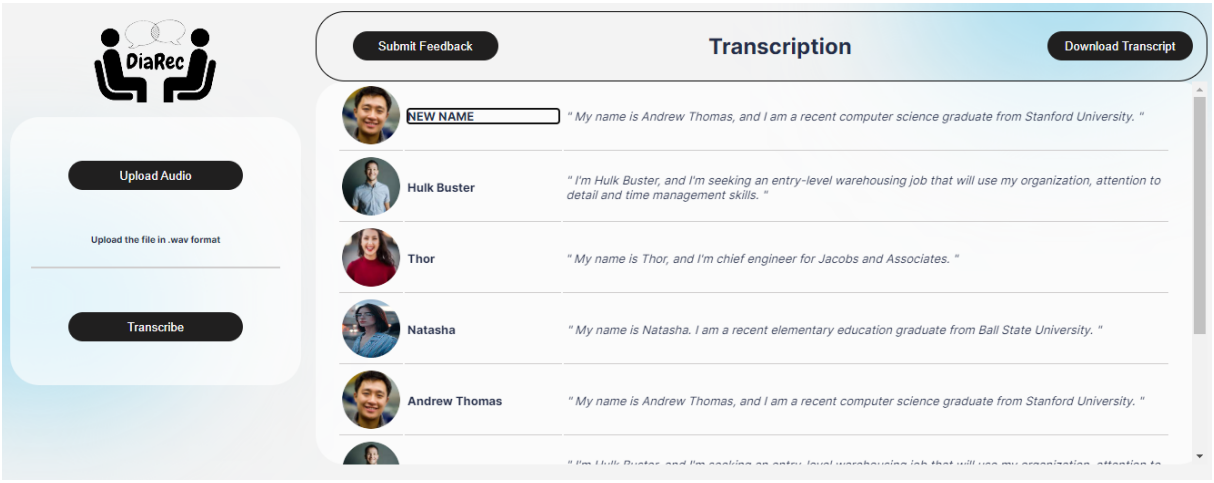

Figure 1 *Proposed Speaker Interface to Collect User Feedback*

In the second step, a semi-supervised learning approach is designed to have a speaker recognition system that can learn from both labeled and unlabeled data. To do this, we will use a graph-based method called label propagation (LP) to infer pseudo-labels for the unlabeled data. This involves "propagating" the labels from a small set of labeled data to a larger set of unlabeled data, in order to create a pseudo-labeled dataset (see the following figure). The

resulting classification system will be able to make use of the intrinsic structure revealed by both the labeled and unlabeled speaker data.

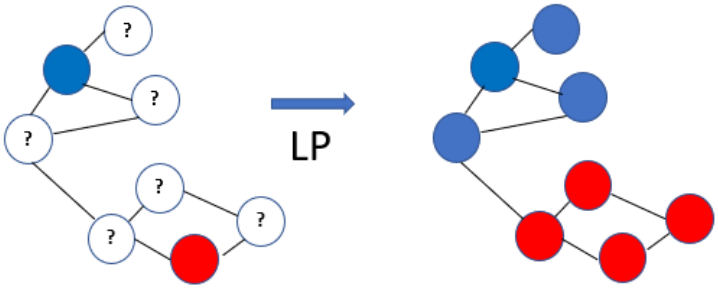

Figure 2 *An Explanation of the Label Propagation (LP) Algorithm*

In the final step, we will incrementally train an existing classifier model to recognize new speaker classes without access to the data that was used to train the original speaker recognition model. This approach will both reduce training costs and address data privacy concerns related to the previously used data.

3. Future Plan

The programs released this year by the SELMA project are still research (and so very recent) tools with very nice results in terms of accuracy, covering the SELMA WP3 tasks. Some of them, like the TTS software, were mature enough to be integrated into the SELMA platforms.

Starting from the first success on the injection of textual data into the training of speech-to-text end-to-end models, the automatic post-editing task will be first addressed as an “on-the-fly“ fine-tuning approach of the implicit language model embedded in the end-to-end neural architecture. In addition, the user feedback entered into the GUI will also be considered and coupled with entity information provided by T2.1 and T2.2.

Another approach that we are considering is the extension of the use of translation memory like proposed in [Khandelwal et al., 2021][1] where k-nearest-neighbor machine translation (kNN-MT) predicts tokens with the nearest neighbor classifier over a large datastore of cached examples (see Figure 3). In our scenario, we would use the user feedback as cached examples and extend this approach to automatic transcription, named entity recognition, and machine translation.

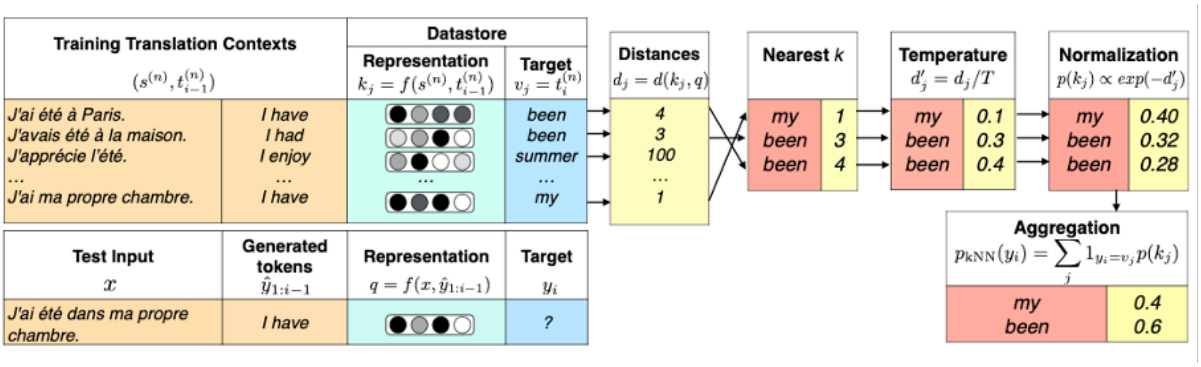

Figure 3 *Computation of the kNN Distribution*

In the upcoming months, the other programs will be packaged and profiled to be integrated into these platforms. At the same time, efforts will be made to extend the language coverage to match the SELMA objectives.

[1] Khandelwal et al., 2021, Nearest neighbor machine translation, International Conference on Learning Representations (ICLR 2021), https://arxiv.org/pdf/2010.00710.pdf