SmarTerp: A CAI System to Support Simultaneous Interpreters in Real-Time *

Susana Rodríguez[1], Roberto Gretter[2], Marco Matassoni[2], Daniele Falavigna[2], Álvaro Alonso[3], Oscar Corcho[3], and Mariano Rico[3]

[1] Independent researcher, Madrid, Spain
[2] Fondazione Bruno Kessler, Trento, Italy
[3] Universidad Politécnica de Madrid, Madrid, Spain

Abstract. We present a system to support simultaneous interpreting in specific domains. The system is being developed thanks to a strong synergy among technicians, mostly experts on both speech and text processing, and end-users, i.e. professional interpreters who define the requirements and will test the final solution. Some preliminary encouraging results have been achieved on benchmark tests collected with the aim of measuring the performance of single components of the whole system, namely: automatic speech recognition (ASR) and named entity recognition.

Keywords: Computer-Assisted Interpretation · Multilingual Knowledge Graphs · Automatic Speech Recognition

1 Introduction

Simultaneous interpreting is a very cognitively demanding task consisting in the execution of different processing sub-tasks in parallel. As an example, if we take the interpretation of numbers, a high error or omission rate is observed, especially in the case of interpreters working in isolation (without a booth-mate, as in remote simultaneous interpreting or RSI), ranging from 70% in the case of students to as much as 40% in the case of professional interpreters [2]. As a further example, a study reported in [9] shows that the number of disfluencies (i.e. hesitations) produced by interpreters is significantly higher than that produced by non interpreters, mainly due to the lexical richness of interpreters themselves. The SmarTerp project aims to develop a Computer-Assisted Interpretation (CAI) system to support the simultaneous interpreter, especially in the RSI modality, by addressing the entire workflow of the interpreting activity, from the preparation of specialised multilingual glossaries that will serve to feed and train the ASR and AI built into the system and extract and propose terminology (e.g. named entities, numerals, etc) to assist the interpreter in real-time, to the post-event validation of new entries by the interpreter that will be fed back into the system to perpetuate a virtuous circle of generating and accumulating specialised knowledge for recurrent use by the interpreter/team of interpreters and the end-customer of the interpreting services.

Although the proposal of using both natural language processing (NLP) and ASR technologies is not new for developing CAI tools (see e.g. the works reported in [3] for a good review) and, at the same time, there are projects, such

* Under the aegis of the EIT Digital, supported by the European Institute of Innovation and Technology (EIT), a body of the EU

Translation and Interpreting Technology Online, pages 102–109,
July 05-07, 2021.
https://doi.org/10.26615/978-954-452-071-7_012

as EABM[1] that aim to use extensively ASR technology to create user-friendly interpreting interfaces, we believe that the strong synergistic effort produced in the SmarTerp project among NLP/ASR experts, software developers and end-users, aimed at both defining the requirements and evaluating and refining the performance of the resulting CAI system, can provide a significant step forward in the development of such tools.

2 Automatic Transcription of Audio

One of the requirements of the ASR systems used in the SmarTerp project is that they have to perform well on specific application domains. More precisely, the source language to be translated by the interpreter may contain a large number of technical terms and morphological variations that are usually not present (or occur with low frequencies) in "general purpose" training text corpora. The result is that a general purpose language model (LM) exhibits on in-domain data high values of both out-of-vocabulary (OOV) word rates and perplexities, worsening the word error rate (WER) of the ASR system that utilises it. To alleviate this effect we propose a procedure, described in section 2.2, that extracts from a given corpus the texts that are "closest", in some way, to a glossary of terms furnished by an interpreter. This one is assumed to contain most of the important words of the subject of a given interpretation session. Then, taking advantage from previous experience for estimating the proficiency of second language learners (see [6]), we developed a procedure, summarised in section 2.2, to adapt a general purpose LM to the domain of each interpretation session. This way we are able to instantiate an ASR engine specific to each interpretation session. Note that this has a strong impact on the whole architecture of the system, since it requires to update, on demand by the interpreters, the LM of each ASR engine.

2.1 Acoustic Models

The acoustic models are trained on data coming from CommonVoice [1] and Euronews transcriptions [7] , using a (Kaldi) standard *chain* recipe based on lattice-free maximum mutual information (LF-MMI) optimisation criterion [8]. In order to be more robust against possible variations in the speaking rate of the speakers, the usual *data augmentation* technique for the SmarTerp models has been expanded, generating time-stretched versions of the original training set (with factors 0.8 and 1.2, besides the standard factors 0.9 and 1.1).

Table 1 summarises the characteristics of the audio data used for the models in our five working languages.

2.2 Language Models

As previously mentioned, we assume a glossary will be available from which to derive some *seed words* that will be used, in turn, both to update the dictionary of the ASR system and to select LM adaptation texts from the available training corpora. These ones are derived both from Internet news, collected from about 2000 to 2020, and from a Wikipedia dump. Table 2 reports some statistics related

[1] see https://www.eabm.ugent.be/EABM

Table 1. Audio corpora for training the acoustic models.

Language	CV (h:m)	EuroNews (h:m)	Total Speakers	Running words
English	781:47	68:56	35k	5,742k
French	432:07	59:42	14k	3,637k
German	426:30	70:47	13k	3,196k
Italian	148:40	74:22	9k	1,727k
Spanish	322:00	73:40	16k	2,857k

Table 2. Text corpora for training the LMs for ASR in SmarTerp. Mw means millions of running words.

Language	Lexicon size	Total running words	Internet News	Wikipedia 2018
English	9,512,829	3790,55 Mw	1409,91 Mw	2380,64 Mw
French	4,422,428	1442,85 Mw	536,06 Mw	906,79 Mw
German	8,767,970	2015,47 Mw	972,89 Mw	1042,58 Mw
Italian	4,943,488	3083,54 Mw	2458,08 Mw	625,46 Mw
Spanish	4,182,225	2246,07 Mw	1544,51 Mw	701,56 Mw

to the training corpora used in this work for 5 different languages. Note that the huge lexicon size is due to the fact that Internet data have a very long queue of questionable terms (typos, etc.). To accomplish the task of text selection we implement the following steps:

- selection of the **seed words**, i.e. technical words that characterise the topic (i.e. the interpretation session) to be addressed; they are simply the words, in the glossary provided by the interpreter, that are not in the initial lexicon (composed by the most frequent 128 Kwords of that language);
- selection of the **adaptation text**, i.e. sentences in the training corpus that contain at least one of the seed words. Note that we hypothesise not having additional texts related to the topic to be addressed;
- creation of both the **adapted lexicon** and **adapted LM**.

Since several approaches can be employed to obtain and use the seed words (e.g. based on texts' distance, texts' semantic similarity, etc) we define the following indicators that allow to measure their effectiveness on benchmark tests (see section 5) collected and manually transcribed within the SmarTerp project.

- OOV rate. Since OOV words cannot be part of the ASR output, they will certainly be errors. We try to get a low OOV rate without increasing too much the lexicon size.
- WER of the ASR system.
- Precision, Recall and F-measure on a subset of technically significant words (hereafter called important words), manually marked in the benchmarks.

3 Semantic Interpretation

Once the transcripts are generated from the audio input, the role of the semantic interpretation module is to detect relevant entities that appear on these transcripts and that may be of interest for the interpreters. Examples of such entities

are those that may be difficult for them to translate during the interpretation session, such as terms that are very specific to the domain or numerical values, which are known to be hard to translate since they require an additional cognitive effort due to the transcoding exertion they require, etc.

The main challenge in this context is that we are not dealing with a typical Named Entity Recognition problem, where elements like persons, organisations, places, etc., need to be detected. That is, recognising the entity "United States" in the text and offering its potential translation into Spanish "Estados Unidos" may not make much sense in the context of the whole system, since this is commonly a well-known term for interpreters. Using an example of the dentistry domain, it is rather more useful for an interpreter to identify the noun "flap" and provide its translation into Spanish ("colgajo"), or to identify a numerical value ("nineteen_seventy_six") and transform it into the Arabic numeral (the year 1976) the interpreter will recognise and introduce in the interpreted speech (in the target language) with little or no effort. Therefore, we need to talk about Interpreter-relevant Term Recognition and their translation into the target language.

To perform this type of task, the module is based on the usage of a layered set of multilingual general purpose, domain-specific and user-specific knowledge graphs, following best practices in the representation of multilingual linked data, as described in Section 3.1. The translation of numerical entities is discussed in Section 3.2.

3.1 Multilingual Knowledge Graphs

The terms and entities that are used by the system are represented using common practices for multilingual Linked Data [5]. These ensure that given an entity or term identified in the knowledge graph (e.g., https://www.wikidata.org/wiki/Q30 for Wikidata's term for the United States of America), the labels in different languages would be easily available using simple SPARQL queries.

As discussed in Section 2, for the overall system to work adequately it is important to adapt the underlying resources (in the case of this module, the multilingual knowledge graphs) to the interpreting sessions that are going to be performed. In our case, the resource management strategy of the multilingual terminologies that are used by this module differs slightly from the approach followed for the adaptation of language models used in the ASR component.

Instead of adapting a single resource, in our case we maintain three layers of multilingual knowledge graphs, which are used as the basis for the identification of terms to be translated and presented to the interpreter. The first layer contains rather small knowledge graphs that are generated from the multilingual glossaries (dictionaries) that are commonly maintained by interpreters, with domain-specific and event-specific terms. These glossaries are commonly edited by interpreters using spreadsheets, where each column contains terms, acronyms, etc., in every language of interest, and are commonly used by them when working on an interpreting session. The second layer contains domain-specific knowledge graphs (e.g., from the medical domain) that are generated from publicly available resources. This second layer is activated after the first one, when there are potentially-relevant terms that have not been identified in the first layer. The final layer contains an extract of existing knowledge graphs

like Wikidata and DBpedia (with the relevant languages used during the interpretation session and only containing term URIs and their labels in different languages) that can be used in case that none of the previous ones are activated. In order to provide a very fast access to these multilingual knowledge graphs with a low memory consumption, we have generated Header-Dictionary-Triples (HDT) versions of them [4]. For each group of n words coming from the transcripts, we obtain the tokens and use combinations of 1 to 5 n-grams (words) so as to look for these terms in the different layers. Although this may seem like a brute force approach, our initial experiments have shown that it allows identifying relevant terms in the generated transcripts.

3.2 Numerical Entity Translation

Numbers are identified in the transcriptions provided by the ASR system using a special notation with underscores (e.g., `_sixty nine_`). This allows the semantic interpretation system to identify these terms easily, so that they do not need to be submitted to the knowledge-graph-based structure that was presented in the previous section. The transformations for numerical entities are implemented following a simple rule-based approach where the typical types of transformation across languages have been identified by a group of interpreters (transformations for years across languages, transformations for units used in quantities, etc.)

4 System Architecture

Fig. 1 shows the block diagram of the integration between the following modules:

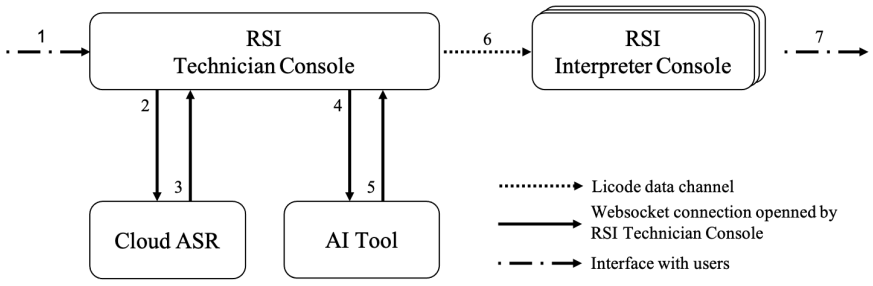

Fig. 1. Block diagram of RSI - ASR - AI integration

– **RSI Technician Console** represents the web-based interface used by the technical staff managing an interpreting session. Thanks to this interface, the technician is able to introduce in the system the audio and video flows of the conference speakers. On the other hand, **RSI Interpreter Console** represents the set of software consoles used by interpreters to visualise both the speakers' video and the materials (e.g. presentation slides) shared by them as well as to receive their audio. Interpreters also use these consoles to manage the input and output audio channels for developing the necessary tasks during an interpreting session. They can switch the input channels

from the technician one (i.e. floor) to the ones shared by other interpreters and modify their output language channel. In this console interpreters also see the AI tool output as explained below.
- **Cloud ASR** represents the on-demand cloud service in charge of transcribing in real-time an audio input in different languages for producing a text output. Its interface supports the receipt of a set of consecutive audio chunks (having a duration that will be optimised during the test phase of the SmartErp project) extracted from an audio flow through a websocket. As a result the Cloud ASR system sequentially respond with the text transcription in the same websocket connection using a JSON document.
- **AI Tool** receives an audio transcription and generates a set of terms for helping interpreters to perform their job. Its interface support the receipt of a set of consecutive JSON documents with a transcription through a websocket. As a result the AI tool generates the terms and sends them through the websocket interface when ready.

The complete interaction flow between the modules can be summarised as follows: 1) Using the JavaScript MediaStream API, the interface asks permission to access web camera and microphone of the PC used by the technician. This generates a video stream and an audio stream. These streams are shared with the interpreters using Licode [2], an open source multi videoconferencing platform based on WebRTC. 2) Thanks to the AudioContext API, the interface extracts audio chunks from the audio stream and sends them to the Cloud ASR using a previously opened websocket. 3) Cloud ASR synchronously answers with the text transcription using the format described above. 4) After receiving the transcription, the technician interface sends each JSON object with the transcription to the AI Tool using a second websocket connection previously created. 5) The AI Tool process the transcriptions and asynchronously generates the terms for interpreters. These terms are sent to the RSI technician console through the websocket connection. 6) Using Licode data channel the technician console multicasts the terms generated by the AI tool to the consoles of the interpreters connected to the same session. These terms are displayed, without a significant delay, to the interpreters in the console. 7) Audio output in the different available languages is sent to the assistants of the conference and to other interpreters in the same session.

5 System Evaluation

As mentioned above, in SmartTerp we prepared benchmarks for the 3 languages of the project (English, Italian, Spanish) plus two important European languages, French and German. Table 3 reports duration and number of words of the benchmarks; French and German are still in a processing stage. Data were collected and manually transcribed using Transcriber[3], a tool for segmenting, labelling and transcribing speech. In addition to time markers and orthographic transcription of the audio data, we decided to label with parenthesis Important Words (IWs), which represent content words that are significant for the selected domain (i.e. dentistry) and are a fundamental part of the desired output of the automatic system.

[2] https://lynckia.com/licode
[3] http://trans.sourceforge.net/

Table 3. Benchmarks collected and annotated in SmarTerp.

language	recordings	raw duration	transcribed duration	running words	running IWs
English	5	04:02:34	03:03:06	28279	3343
French	12	03:22:07	–	–	–
German	∼16	∼03:00:00	–	–	–
Italian	33	05:29:34	04:10:31	31001	4560
Spanish	13	03:09:53	03:01:59	25339	3351

Preliminary ASR results on the completed benchmarks are reported in Table 4, with and without the adaptation stage. Together with OOV rate and lexicon size, we report WER computed on all the uttered words (including functional words, which are useless for this task), and precision/recall computed only on IWs that, since they represent the most technically significant words in the domain, are more related to the output desired by interpreters. It is worth noting that the adaptation system is effective for all of the three languages and for all the considered metrics. Low WER for English is partly due to a scarce audio quality in the recordings, that mainly affects functional words: this explains the English high precision, which is computed on IWs only.

Table 4. Preliminary results for baseline and adapted systems. Both WER on all words and precision/recall/F-measure on isolated IWs are reported.

language	Lexicon size	OOV rate	WER	IWs: P / R / F
English baseline	128041	1.93%	26.39%	0.96 / 0.59 / 0.73
English adapted	213237	0.79%	23.34%	0.97 / 0.71 / 0.82
Italian baseline	128009	3.51%	15.14%	0.95 / 0.67 / 0.79
Italian adapted	1197995	1.02%	11.73%	0.98 / 0.82 / 0.90
Spanish baseline	128229	4.09%	22.60%	0.93 / 0.56 / 0.69
Spanish adapted	236716	1.14%	17.74%	0.98 / 0.75 / 0.85

6 Conclusions

The SmarTerp project is an on-going innovation action funded by the EIT Digital aiming to develop a Computer-Assisted Interpretation system to support the cognitively demanding task of simultaneous interpretation with state-of-the-art language technology. To do so, the consortium, created to solve the many challenges the real-time constraints impose on the system, has obtained so far encouraging results. In particular: *a)* good performance on specific application domains by the ASR systems thanks to a procedure that extracts from a given corpus the texts that are closest to a typical interpreters' glossary and adapts a general purpose LM to the domain of each interpretation session; and *b)* devising of a semantic interpretation module to detect relevant entities that appear on the ASR transcripts and that may be of interest for the interpreters, such as named entities and terms that are very specific to the domain, or numerical values, which are known to be difficult to interpret since they require an additional cognitive effort due to the transcoding exertion they require.

References

1. Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., Weber, G.: Common voice: A massively-multilingual speech corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 4218–4222. European Language Resources Association, Marseille, France (May 2020)
2. Desmet, B., Vandierendonck, M., Defrancq, B.: Simultaneous interpretation of numbers and the impact of technological support. In: Multilingual Natural Language Processing. pp. 13—27. No. 11, C. Fantinuoli ed. Berlin: Language Science Press (2018)
3. Fantinuoli, C.: Interpreting and technology. In: Multilingual Natural Language Processing. No. 11, C. Fantinuoli ed. Berlin: Language Science Press (2018)
4. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary rdf representation for publication and exchange (hdt). Web Semantics: Science, Services and Agents on the World Wide Web **19**, 22–41 (2013), http://www.websemanticsjournal.org/index.php/ps/article/view/328
5. Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P., McCrae, J.: Challenges for the multilingual web of data. Journal of Web Semantics **11**, 63–71 (2012)
6. Gretter, R., Matassoni, M., Allgaier, K., Tchistiakova, S., Falavigna, D.: Automatic assessment of spoken language proficiency of non-native children. In: Proc. of ICASSP (2019)
7. Gretter, R.: Euronews: a multilingual speech corpus for ASR. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 2635–2638. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014)
8. Manohar, V., Hadian, H., Povey, D., Khudanpur, S.: Semi-supervised training of acoustic models using lattice-free MMI. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 4844–4848 (2018)
9. Plevoets, K., Defrancq, B.: The effect of informational load on disfluencies in interpreting: A corpus-based regression analysis. The Journal of the American Translation and Interpreting Studies Association **11**(2), 202–224 (2016)