xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.3

Spoken and written language analysis techniques

Dissemination level:	Public
Contractual date of delivery:	Month 12, 31 October 2021
Actual date of delivery:	Month 13, 1 November 2021
Work package:	WP3 - Analysis and fusion of multi-modal data
Task:	T3.3 – Multilingual audio and written language analysis
Type:	Demonstrator
Approval Status:	Final version
Version:	1.0
Number of pages:	27
Filename:	D3.3_xR4Drama_LanguageAnalysisTechniquesV1_ 20211101_v1.0.pdf

Abstract

This deliverable describes the initial versions and outcomes of the audio and written language analysis components of xR4DRAMA developed in T3.3 of WP3. This component is responsible for (i) the transcription of spoken language into text and (ii) the analysis of textual and spoken (i.e., speech data transcriptions) material obtained from different sources, including communication data from citizens, textual information provided by location scouts, social media messages and online information.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	06-09-2021	Table of contents	Montserrat Marimon
0.2	22-10-2021	First draft	Montserrat Marimon
0.3	27-10-2021	Draft for Internal Review by CERTH	Montserrat Marimon
1.0	01-11-2021	Final version	Montserrat Marimon

Author list

Organization	Name	Contact Information
UPF	Mónica Dominguez	monica.dominguez@upf.edu
UPF	Montserrat Marimon	montserrat.marimon@upf.edu
UPF	Alexander Shvets	alexander.shvets@upf.edu
CERTH	Sotiris Diplaris	diplaris@iti.gr
CERTH	Anastasios Karakostas	akarakos@iti.gr

Executive Summary

This deliverable describes the initial versions and outcomes of the audio and written language analysis components of xR4DRAMA developed in T3.3 of WP3. This component is responsible for (i) the transcription of spoken language into text and (ii) the analysis of textual and spoken (i.e., speech data transcriptions) material obtained from different sources, including communication data from citizens, textual information provided by location scouts, social media messages and online information. We describe technical details and report the results of initial evaluations.

Abbreviations and Acronyms

AM	Acoustic Model
ASR	Automatic Speech Recognition
CE	Concept Extraction
CNN	Convolutional Neural Network
DSynt	Deep Syntactic
GPU	Graphics Processing Unit
HMM	Hidden Markov Models
LM	Language Model
LSTM	Long Short-Term Memory
MFCC	Mel Frequency Cepstral Coefficient
NER	Named Entity Recognition
RAM	Random Access Memory
RNN	Recurrent Neural Network
SSynt	Surface Syntactic
UD	Universal Dependencies
VR	Virtual Reality
WER	Word Error Rate

List of Figures

List of Tables

Table of Contents

1 INTRODUCTION

Multilingual audio and written language analysis are integrated in the xR4DRAMA platform to analyse the material acquired in T2.1 or provided by the users from several sources for the derivation of abstract linguistic representations, which can be used by the language generation component that will export information relevant to the needs and requirements of the users. This deliverable describes the initial version of the techniques and methodologies for the linguistic analysis in the project.

In Section 2 we provide an overview of state-of-the-art, off-the-shelf techniques in automatic speech recognition and discuss advantages and drawbacks of certain approaches. We report on the results of the evaluation carried out for PUC1 on Disaster Management in Italian and PUC2 on Media Production Planning in English using the Wav2Leter++, the chosen architecture, for ASR.

In Section 3 we describe the different components included in the language analysis pipeline; that is, the concept extraction (subsection 3.1) and the named entity recognition (subsection 3.2) components, which serve to detect and classify linguistic expressions that indicate relevant entities; the temporal expression identification component (subsection 3.3), which extracts and normalizes temporal expressions; the word sense disambiguation (subsection 3.4) and the geolocation (subsection 3.5) modules, which annotate input texts with references to lexical meanings in databases, while also providing the respective geographic coordinates, where applicable (i.e., location-type entities, such as “Vicenza”), the surface language analysis component (subsection 3.6), which covers the linguistic analysis from tokenization up to surface syntax (dependency) parsing and finally, the deep analysis component (subsection 3.7) which covers the semantic analysis and generates structured representations that will be stored in the Knowledge Base.

Section 4 concludes the deliverable.

2 AUTOMATIC SPEECH RECOGNITION

2.1 Task Definition and Related Work Summary

Automatic Speech Recognition (ASR) addresses the task of transcribing natural spoken language into text. This area of speech recognition technologies started in the second half of the 20th century with the development of simple applications that could recognize isolated digits and words. For example, Audrey system, built by Bell Labs in 1952, is considered to be the first speech recognition device and was able to recognise only ten digits spoken by a single voice. Another example of such a simple ASR application was Shoebox, from IBM, which was built in 1961 and could recognize 16 isolated words and perform mathematical computations[1].

Research in ASR has traditionally addressed different tasks that vary in difficulty: digit recognition, word recognition, keyword spotting, language identification, command recognition, sentence recognition, and at the top of the list, continuous speech transcription. To illustrate the challenge in continuous speech transcription, it is worth mentioning that even commercial systems, e.g. Google's Cloud speech-to-text, report processing quotas of 1 minute for synchronous requests and 5 minutes for streaming requests[2].

The challenge of automatically recognizing speech is composed of two steps: i) recognizing phonemes (the minimal unit of spoken language that can distinguish one word from another, e.g. bat and bet) pronounced by different speakers in different phonemic contexts; and ii) modelling all the possible combinations of words in a given language, which might not be infinite (as not all combinations are grammatically possible), but they are indeed numerous (the way we communicate a message can also vary depending on the context, domain, speaker, etc.). Two modules in ASR systems handle these two prediction problems: the acoustic model and the language model.

In the last decade, the problem of transcribing whole sentences has been solved to more or less reasonable extent thanks to: (i) advances in classification algorithms and therefore, computational power to handle large amounts of data, and (ii) design of corpora to train these systems to tackle well-defined domains, that is, reducing the illimited (possibly infinite) number of possible word combinations to a specific data so that the amount of possible utterances is more manageable. However, escalating systems to be able to recognize any speaker in any language talking non-stop for an indefinite amount of time is still a problem that remains unsolved.

A clear distinction must be made between industry and academia solutions in the field of ASR. Commercial systems (commercial applications implemented in the industry) are using the latest advances in machine learning together with an enormous amount of training data. An example can be found in the Parrotron system by Google that uses only for the main ASR module a ~30,000 hour training set consisting of about 24 million English utterances (Biadsy, F. et al. 2019). Open-source applications (implemented within academic contexts) are usually trained with a much more limited amount of data, which directly impacts on their

[1] https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html
[2] https://cloud.google.com/speech-to-text/quotas

performance assessed in terms of Word Error Rate (WER). Whereas commercial systems usually perform at under 10% WER for standard tasks in well-resourced languages, open source systems perform at about 15% WER in standard English on widely used corpora such as TEDLIUM (Rousseau, A. et al. 2012) or Switchboard (Godfrey, J. et al., 1992), they go down to 35% WER in distant or noisy tasks and 55% WER in new domains or under-resourced languages (Yashesh, G. et al., 2016). It must be noted that humans transcribing speech usually get around 18% WER for challenging tasks in English and French dealing with homophones (words that sound the same but have a different spelling); cf. (Vasilescu, I. et al. 2011). The technology behind this might be quite similar, but there are two key factors that make the quantitative gap bigger: the computational power needed to train complex neural networks and the access and processing of large amounts of data. A recent study by Iancu (2019) analyzed the performance of Google's ASR system in a low resource language using videos on different subjects for e-learning and showed that even this commercial system was able to reach a modest performance of 31% WER for this task.

Finally, it must be noted that the problem of computing WER without having a standard benchmark to compare to, makes it difficult to really assess these systems. The cause of this is the lack of a common ground scenario that is agreed upon to tackle the most important challenges from a linguistic point of view. A recent study by Béchet (2019) shows that the corpora used in training and evaluation of Spoken Language Understanding performance actually address the commonest and simplest linguistic problems and leave aside the real complex ones.

2.2 Selection of an off-the-shelf ASR application

2.2.1 Assessment of different technologies

The main motivation to use an open-source system is to comply with the security protocols for data established within the context of the project, as speech is considered sensitive data. Among the ASR frameworks available under an open-source license, the following applications were initially considered:

- KALDI **https://kaldi-asr.org/**: Kaldi is a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0. Kaldi implements the use of neural networks which have been proved as the state-of-the-art technology in the field of ASR. The toolkit includes recipes for most languages and widely used corpora. However, Kaldi is intended for use by speech recognition researchers mainly, and it does not include a frontend to capture the speech signal and convert it to digital form.
- HTK **http://htk.eng.cam.ac.uk/**: The HTK toolkit has been implemented for Mandarin conversational telephone transcription tasks. Recipes for other languages are not available and the documentation is scarce.
- CMU Sphinx **https://cmusphinx.github.io/**: The CMU Sphinx toolkit is a leading speech recognition toolkit with various tools used to build speech applications. CMU Sphinx contains several packages for different tasks and applications. Pretrained models for many languages are available (including several dialects of English and Spanish, German, Greek, Portuguese, Dutch, French, Russian, Italian, Catalan, , Hindi, Mandarin, Kazakh). Available tools include frontend and backend that can be easily

integrated via a Java environment. Detailed documentation and a step-by-step tutorial are available online[3].

- RWTH ASR https://www-i6.informatik.rwth-aachen.de/rwth-asr/: So far, this toolkit has been developed only for English and Spanish and it is trained on EU parliament transcriptions[4].

Our initial idea was using KALDI, as the literature reports this system to be state-of-the-art in the field of ASR together with CMU Sphinx. The main issue we had when trying to deploy Kaldi was the fact that the system has been designed for research purposes, therefore it does not have an inbuilt frontend that can take an audio input and convert it to digital form. We found an open-source interface that could be integrated as frontend for Kaldi called Eesen transcriber[5], which is built using a virtual machine that runs either locally with Vagrant/VirtualBox or remotely as an Amazon Machine image on AWS. However, after trying to integrate the local virtual machine, we estimated that the amount of work would exceed the initial PMs. Moreover, the Eesen transcriber has been designed to perform a different task (i.e., keyword spotting) from the one intended in xR4DRAMA (i.e., transcribing continuous speech). It was intended to provide an online video browser service using keywords, also known as keyword spotting. As mentioned above, the task of keyword spotting in ASR is a much simpler task than continuous speech transcription, that is why we discarded the use of both Easen transcriber and Kaldi.

Recent advances in machine learning have shown that neural network architectures for ASR yield a much more improved performance on WER than traditional architectures based on Hidden Markov Models (HMMs). The counterpoint to such a better performance is that these neural architectures usually require much larger amounts of training data (above one thousand hours of speech and 1.000.000 sentences) as well as a demanding hardware infrastructure (multiple GPUs) to run the training (cf., e.g., Hannun, et al. 2014). However, steady efforts have been made in the field of speech technologies to promote the collection and sharing of large speech corpora, like the crowdsourced initiative Common Voice dataset[6] and the Multilingual LibriSpeech Corpus[7]. On the other hand, lighter architectures have been released with open-source pre-trained models, such as Wav2Letter++ (Pratap, et al. 2019) based on Flashlight[8] and Array Fire Tensor[9] libraries.

The Wav2Letter++ is an open-source speech processing toolkit written in C++ and developed by the Facebook AI Research team. Previous state-of-the-art neural speech recognition systems were built on recursive neural networks (RNNs) for acoustic or language modeling. The Wav2Letter++ architecture allows an alternative approach based on convolutional

[3] https://cmusphinx.github.io/wiki/
[4] http://catalog.elra.info/en-us/repository/browse/ELRA-S0250/
[5] https://github.com/srvk/eesen-transcriber
[6] https://commonvoice.mozilla.org/en/datasets
[7] http://www.openslr.org/94/
[8] A C++ library for machine learning. https://github.com/facebookresearch/flashlight
[9] https://github.com/arrayfire/arrayfire.git

neural networks (CNNs)[10]. CNNs have been proved to yield a much better performance in image detection tasks due to their specialization of learning patterns at different levels, thus being able to distinguish edges of shapes in images, and thus spotting larger areas and image contours. The neural network architecture applied to speech recognition, eliminates the feature extraction step of mel-frequency cepstral coefficients (MFCCs) as it is trained end-to-end to predict characters from the raw waveform. This advancement eradicates the need of word aligned transcriptions to train the acoustic model. The Wav2Letter++ architecture comprises an acoustic model, a lexicon (similar to the dictionary in the CMU Sphinx toolkit) and a language model to decode words.

2.2.2 Dockerization of Wav2Letter++

A Docker image with the latest version of the Wav2Letter++ architecture has been built at UPF for the integration in the xR4DRAMA platform and the facilitation communication between components. We also provide a callable online service with our ASR component that will take audio input in all the languages of the project (i.e., English, German, and Italian). The component includes a converter of audio files to be processed by the ASR application that can take the commonest audio formats: mp3, wav, ogg, aiff, etc[11].

With respect to open-source speech and language resources to train neural architectures, great advances have been made. As neural architectures require a considerably larger amount of data for training, open-source resources have become essential. Multilingual corpora, pre-trained acoustic and language models for Wav2Letter are presented in the work by Pratap et al. (2020) and made available in OpenSLR.org[12]. A major concern that needs to be raised now is storage and processing capabilities for these resources. Table 1 shows the sizes of the resources in Patrap et al. (2020), including acoustic corpora, pre-trained acoustic model (AM), and language model (LM). It must be noted though that LMs are made available in arpa format that can be converted to a lighter binary file. In the evaluation of Wav2Letter++ presented in the next section, the RASR model[13] was used for English and the Multilingual Librispeech (MLS)[14] model is available in German and Italian.

Table 1: Resources' size of the Multilingual LibriSpeech (MLS) dataset and models.

Language	Corpora	AM (binary)	LM (arpa)
English	2.4T	1G	44.0G
German	115.0G	1G	2.7G
Italian	15G	1G	1.7G

[10] It is out of scope for this deliverable to provide a full explanation on deep learning architectures. Nevertheless, a detailed description can be found in (Goodfellow, et al. 2016) **https://www.deeplearningbook.org/**

[11] See **http://sox.sourceforge.net/soxformat.html** for an exhaustive list of audio formats

[12] **http://www.openslr.org/94/**

[13] **https://github.com/facebookresearch/wav2letter/tree/master/recipes/rasr**

[14] **https://github.com/facebookresearch/wav2letter/tree/master/recipes/mls**

Preliminary tests on the xR4DRAMA domain were run locally in a PC with an Intel(r) Core (TM) i7-3632QM processor, 2.20GHz CPU and 16GB of RAM under Ubuntu 18.04 OS on a partition with 40GB of disk space. Even though the system had a reasonable response time especially in English, Italian and German models were a bit slower, so we created a binary LM for these languages, which dramatically increased response time. For the integration in the xR4DRAMA platform, we will use the dockerized image of the Wav2Letter++ architecture that will need to access the AM and LM of the different languages ideally stored in a dedicated repository.

In the next section, we will report on the results of the evaluation carried out for PUC1 on Disaster Management in Italian and PUC2 on Media Production Planning in English using the Wav2Leter++ architecture for Automatic Speech Recognition. So far, German has not been tested in the xR4DRAMA domains, as it is not clear yet whether the user partners are interested in this functionality. We did some testing ourselves with out of domain read speech from native speakers of German and the results are presented below.

2.2.3 Evaluation of Wav2Letter++ in the languages of xR4DRAMA

The main objective of the Wav2Leter++ ASR architecture's evaluation is to test pre-trained models for English, German, and Italian on the use case scenarios foreseen for ASR within the context of xR4DRAMA (except for German that is tested on general read speech).

The English test set consists of three YouTube documentary videos provided by the user partner Deutsche Welle (DW) with the following titles and total lengths:

- Trying To Live On Mars For 22 Days[15] (referred as Sample 1 from now on) 6 minutes 51 seconds.
- Trying The Most Disgusting Food At The Disgusting Food Museum Malmö[16] (Sample 2 from now on) lasting 10 minutes.
- Climbing 700 m Above The Abyss: Stairway To Heaven In Austria[17] (Sample 3 from now on) 6 minutes 32 seconds.

All videos included two male voices: a reporter (same speaker in all three documentaries) and an expert in the field of each documentary (namely, an astronaut, the director of the museum and a climber). The original audio file included background music and different microphone recording environments, studio off-voice for the narrated parts, outside microphone capture, and inside microphone capture including in-helmet microphone recording. None of the speakers were native English speakers, so this makes for a good testing scenario for accented speech recognition by the ASR module which is pre-trained on native read speech in English.

The DW videos were converted to audio format and the music and speech were separated using the open-source online service **https://mvsep.com/**. The speech processing software Praat (Boersma 2021) has been used to split the resulting wav file into manageable speech

[15] https://www.youtube.com/watch?v=yPAqTYzzjz8
[16] https://www.youtube.com/watch?v=0OaMz0KI4gM&t=7s
[17] https://www.youtube.com/watch?v=aLZx0IfwU9g

units, ideally including only one speaker (some fragments contained overlapping speech, usually two people talking at the same time). An automatic functionality of Praat has been used to split the files that detect voice activity and silences. The automatic segmentation was manually revised to check for consistency and the required initial minimum silence for ASR processing. A total of 224 fragments were processed consisting of 49 fragments for Sample 1, 125 fragments for Sample 2 and 50 fragments for Sample 3. The total duration of net speech added up to a total of 15 minutes 30 seconds. Even though most of the fragments were short in the range (between 1 and 5 seconds), there were some long stretches of continuous speech lasting up to 19 seconds.

The Italian test corpus was provided by the user partner AAWA. It consisted of a collection of 20 telephone recordings simulating emergency situations in the city of Vicenza being reported to an emergency call center. Each recording included two speakers: the operator and the citizen reporting the emergency. All speakers were native Italian speakers. The recording often included strong environmental noise and channel disruptions that made the speech unintelligible to the human ear. The register is spontaneous and under stressing circumstances, which affects the resulting speech with hesitations, fillers, ungrammatical sentences, overlapping speech. Such speech samples pose a great challenge for ASR technologies. These audios were also split in manageable shorter files using Praat and checked manually to respect dialogue turns. Consequently, some fragments include one word replies whereas others contain long spontaneous sentences. The conversations usually contain proper names like Vicenza (a town), street names, fake citizen's names in Italian and fake telephone numbers. A transcript was provided by the user partners for each dialogue. These documents were automatically processed to serve as a gold standard in our evaluation. The text was split into the actual words per turn and sentences were normalized (removing punctuation marks, uppercase letters, changing digits to numbers in letters, etc.).

All 20 files in Italian were processed, but only a selection of 7 samples was used in the evaluation due to the mismatch between the actual words in the recordings and the provided transcriptions. The following dialogues were tested:

- 1_Sottopasso_allagato (referred as 01it from now on)
- 3_Esondazione_bacchiglione (03it from now on)
- 7_allevamento_allagato (07it from now on)
- 10_PontePusterla_esondazione (10it from now on)
- 12_risalita_fognaria (12it from now on)
- 13_mancanza_energia_elettrica (13it from now on)
- 15_alunno_smarrito (15it from now on)

The overall length of the testing corpus in Italian is 35 minutes 12 seconds. Each dialogue included several dialogue turns ranging from 6 to 12 speech fragments with a total number of 63 fragments being processed in the evaluation. Therefore, the overall duration of net speech used in the evaluation, after voice activity detection and splitting was performed, was 7 minutes 31 seconds.

The metric to assess ASR performance is word error rate (WER). The WER is derived from the Levenshtein distance working at the word level. WER takes into consideration the number of substitutions, deletions and insertions divided by the total number of words in the reference sentence expressed as a conspicuous scale from 0 to 1. Consequently, a WER of 0 means

that the ASR transcription matches exactly the gold or reference sentence, whereas a WER of 1 means that none of the words in the ASR output matched the reference sentence. However, this kind of measurement provides no details on the nature of errors, so further work is therefore required to identify what is the main source(s) of error and where to focus any research effort. We have used a python library for the computation of WER called jiwer[18], so that results of the evaluation are easily reproducible and traceable.

The following subsections present the results of the evaluation in each language, namely English, German, and Italian.

a) English

Table 2 shows the overall statistics (in a scale from 0 to 1) of WER computation of fragments for three documentaries in English.

Table 2: WER statistics for the English test corpus (on a continuous scale from 0 to 1).

Sample	Average WER	Std WER	Mode WER	Median WER
Sample 1	0.19	0.20	0	0.14
Sample 2	0.26	0.29	0	0.18
Sample 3	0.25	0.24	0	0.17

The overall performance in the 224 fragments is positive despite that the scores show a standard deviation over 0.20, which underlines a fairly scattered sample of scores. Median values are always below average, which shows the sample is skewed to the left. Moreover, mode is 0 for all samples, which underlines the fact that the most repeated WER scores in fragments are 0 and therefore contain no errors. To be precise, sample 1 contains 14 fragments (i.e., 28%) scoring 0 WER, sample 2 has got 40 fragments (i.e., 32%) and sample 3 has got 8 fragments (i.e., 16%). Thus, a total of 28% of fragments were transcribed as in the reference transcript (with 0 errors).

An average 23% WER (considering all fragments and all samples) is a positive result for ASR tested on semi spontaneous accented speech in the domain of xR4DRAMA for the transcription of documentaries.

b) German

Table 3 shows the word error rate (WER) on the read speech corpus compiled for testing Wav2Letter++ in German. The corpus includes 10 sentences (s01 to s10) read by a total of 8 speakers (Spk 1 to 8), two of whom have a Bavarian accent (1_bav, 2_bav).

[18] https://pypi.org/project/jiwer/

Table 3: WER for the German read speech corpus (from 0 to 1).

Spk	s01	s02	s03	s04	s05	s06	s07	s08	s09	s10	Ave.
1_bav	0.34	0.50	0.47	0.56	0.70	0.64	0.40	0.65	0.87	0.37	0.55
2_bav	0.30	0.50	0.55	0.57	0.16	0.45	0.65	0.70	0.34	0.47	0.47
3	0.10	0.15	0	0.31	0	0	0.07	0.15	0	0.12	0.09
4	0	0.04	0	0.23	0.30	0.24	0.10	0.20	0.14	0.19	0.15
5	0	20	0	0.31	0	0	0.14	0.15	0	0.12	0.10
6	0.10	0.20	0	0.31	0	0	0.07	0.15	0	0.34	0.12
7	0	0.04	0.08	0.12	0.20	0.39	0.10	0.15	0.07	0	0.12
8	0.12	0.04	0.34	0.08	0.30	0.39	0.25	0.15	0.40	0.10	0.22

It is worth splitting the average WER results of the German evaluation into standard German and Bavarian dialect as there is a strong deviation between these groups. Standard German speakers obtain a 0.13 WER on average, whereas speakers of the Bavarian dialect show an average 0.51 WER.

These results confirm that, even though the Bavarian dialect poses a challenge for pre-trained models in German of Wav2Letter++, standard German is fairly well recognized ranging from a minimum of 0.09 WER for one of the speakers in the corpus. We must emphasize that the speech samples used in this test were read speech, that means they perfectly suit the ASR German model that is trained with read speech.

c) Italian

Table 4 shows the overall statistics (in a scale from 0 to 1) of WER computation of fragments for the 7 dialogues in Italian. In this table we are including the minimum WER of the fragments as the mode is sometimes not computed due to the continuous nature of the scale and the lack of repetition of values.

Table 4: WER statistics for the Italian test corpus (on a continuous scale from 0 to 1).

Sample	Average WER	Std WER	Median WER	min WER	Mode WER
01it	0.44	0.29	0.36	0.08	-
03it	0.51	0.25	0.42	0.22	-
07it	0.74	0.28	0.80	0.37	1
10it	0.80	0.24	0.86	0.44	1
12it	0.35	0.29	0.25	0.07	-
13it	0.64	0.23	0.71	0.23	-
15it	0.72	0.31	0.90	0.23	1

Results of the evaluation of Italian show an overall poor performance of 0.60 WER on average for these 7 samples. Only two samples scored below 0.50 WER and the minimum WER was never 0.0 in all 63 fragments considered in our evaluation. The mode is 1 in 3 of the 7 samples, which shows how poor the ASR is performing on these fragments. This is confirmed by median WER since 4 samples out of 7 have their median values above the average, which means that scores are skewed to the right.

After assessing the quality of the recordings provided for testing, we had to discard many of them mainly because of the aforementioned unintelligible speech, background noise and overlapping of voices. We think that in order to have an operating ASR application with this type of speech samples, we should need to look into either audio pre-processing techniques or keyword recognition, instead of full-fledged literal transcription that is hard even for human transcribers in such a scenario.

3 ANALYSIS OF TEXTUAL AND SPOKEN MATERIAL

Language Analysis in xR4DRAMA is a complex task that requires the combination of a large variety of components performing a series of steps, going from low-level linguistic analysis, such as tokenization, through higher levels of linguistic complexity, such as dependency parsing, to the extraction of entity-relation-entity triples. The language analysis module, thus, implements a pipeline of multiple components, described individually in the following subsection, each building upon the output generated by previous analysis steps.

3.1 Concept extraction

Concepts, along with other lexical items, form a basis for understanding the meaning of the input text. Since surface forms of concepts in a text can contain several tokens, it is also important to merge them into separate multi-word tokens to get correct dependency structures at subsequent language analysis steps.

The concept extraction (CE) component incorporates a deep neural network architecture that acts in a machine-translation manner: it translates an input sentence into an artificial sentence that contains only concepts separated by a termination token "*", in the same order they appear in the original sequence of tokens. A pointer-generator model proposed in (See et al., 2017) is chosen as a core of the component, as the pointer mechanism implies the ability to cope with unknown out-of-vocabulary words (unseen during the training of the model), which is crucial for robust universal concept extraction in a real-world application, while the "generator" part implies the ability to adjust vocabulary distribution for selecting the words for "translation" based on the whole utterance by reading it right-to-left and left-to-right using bidirectional Long Short-Term Memory (LSTM) units.

To adapt this basic model to the task of concept extraction, we applied several modifications to it: (i) following (Gu et al., 2016), we use separate distributions for copying attention and general attention, instead of one for both; (ii) we work with multi-layer LSTMs for encoders and decoders, as they perform better for this task than single layer ones; (iii) we adapt the forms of input and target sequences to the specifics of the task of concept extraction. The input is composed of tokens and their Part-Of-Speech tags (e.g., "Impressive JJ museum NN with IN free JJ access NN and CC accessible JJ toilet NN"). The target sequence concatenates concepts in the order they appear in the text and separates them by a token "*" especially introduced to partition the output (e.g., "museum * free access * toilet").

Table 5 (*P* stands for precision, *R* - for recall, and F_1 - for F_1-score, i.e., the harmonic mean of the precision and recall) shows the results we achieved for Italian. The Italian test corpus was provided by the user partner AAWA. It consisted of a collection of 20 phone transcripts annotated with 852 concepts in total (143 of them are multi-word (17%)).

The generic open-class CE model showed an overall score of 0.69. Domain adaptation algorithm was developed and applied as a post-processing step in accordance with the results of an error analysis carried out on a part of the dataset. In particular, the algorithm forces the selection of frequent concepts in the domain, discards candidates that were selected due to the common mistakes in the parse output typical for provided phone

transcripts and considers the length of the sentences that are in general shorter than the ones that the open-class model was trained with.

Table 5: Results of evaluation of the CE component

	P	*R*	F_1
Open-class CE model	0.65	0.74	0.69
Open-class CE model + Domain adaptation	0.84	0.87	0.85

3.2 Named Entity Recognition

Detection of names of specific entities is an important step towards understanding the informational contents of the input sentences. While many named entities can be recognised by tools that rely on some database, e.g., entity linking or geolocation components, entities such as people or locations that do not have entries in databases can only be detected using Named Entity Recognition (NER) tools.

NER tagging is done using spaCy 2.0[19], a state-of-the-art entity recognition tool with the default English model. The choice of tool is motivated by the support for multiple languages and abundant documentation. This model annotates the 18 entity types used in the OntoNotes 5.0 corpus (Weischedel et al. 2013), i.e., cardinals, dates, events, facts, geo-political entities, names of languages, law-related names, locations, currency and other financial names, ordinals, organizations, percentages, people, products, quantities, temporal expressions, and names of popular works of art. The annotations can span over single words or multiword expressions.

3.3 Temporal Expression Identification

Identification of temporal expressions, such as times, dates and durations, is also important for the disaster management scenario to know when the reported situation has taken place.

Identification of temporal expressions is done with HeidelTime[20] (Strötgen, 2010), a multilingual, domain-sensitive temporal tagger. It extracts temporal expressions from documents and normalizes them according to the TIMEX3 annotation standard.

3.4 Entity and Word Sense Disambiguation

The linking component annotates named entities and concepts previously detected with disambiguated references to a knowledge base containing lexical units, pairs of form and meaning. Our component uses BabelNet[21] 4.0.1 (Navigli and Ponzetto, 2010), a multilingual knowledge base, as a repository of lexical units, where language-independent meanings are usually referred to as BabelNet synsets. BabelNet covers both named entities and concepts,

[19] http://spacy.oi

[20] **https://github.com/HeidelTime/heideltime**

[21] http://babelnet.org

and maps its synsets to several other knowledge bases, ontologies and lexical resources, most notably language-specific versions of Wikipedia and WordNet.

Disambiguation is approached in the linking component by collecting and ranking candidate meanings of the words in the text transcript (Casamayor, 2021), an approach that involves calculating the salience of a meaning with respect to the whole set of candidate meanings, and its plausibility with respect to the context of its mention in the input texts. Meanings are compared with each other using sense embeddings (Camacho-Collados et al., 2016) and with their context using sentence embeddings calculated from their English Wikipedia and WordNet glosses.

The resulting salience and plausibility scores are used to rank all candidate senses. Deciding between candidate senses of a polysemous word or multiword expression is then done by choosing the one with the highest rank. Whenever two or more expressions with competing interpretations overlap, we assign a meaning to the mention with the highest ranked candidate and discard the meaning for the other mentions. For instance, synsets may be found in BabelNet for “renewable energies”, “renewable”, and “energies”. If the highest ranked synset corresponds to “renewable energies”, then we do not assign meanings to “renewable”, nor to “energies”.

3.5 Geolocation

The two use cases --disaster management and media production planning—in xR4DRAMA require extraction of location mentions in the input sentences. Knowing the spatial coordinates should help to locate the elements at risk in the reported emergency and the available facilities.

UPF makes use of two geographical databases, Open Street Maps and GeoNames, to address this task. Use-case specific search indices are created by pruning the data of the databases according to the region of interest (i.e., Vicenza for PUC1 and Corfu for PUC2) and organising them in memory-efficient structures to reduce the usage of Random Access Memory (RAM).

Identification of location candidates in the input text is based both on the named entity recogniser and on linguistic dependency-based patterns, in synergy with the BabelNet links, to determine whether a mention refers to a location or not. The algorithm to form a search query consists of the following steps: i) if a place-indicating mention, such as “park”, “avenue”, “highway”, etc. is linked via a NAME dependency to a proper name, then their concatenation is marked as a location; ii) if a BabelNet link has been obtained for a single- or multi-word mention, and it includes a reference to a DBpedia entry of the classes dbo:Place or dbo:SpatialThing, then the mention is marked as a location; iii) likewise, if the mention under consideration has been tagged by the NER tool as a location.

3.6 Surface Language Analysis

Surface syntactic analysis predicts the grammatical relations between all the words of the sentence; this includes: segmentation (detection of sentence boundaries, if more than one sentence in the input), tokenization (splitting into words), lemmatization (prediction of the base form), Part-Of-Speech tagging (assignment of grammatical categories) and syntactic

parsing with UD dependency relations, in which one of the related elements is considered the head of the relation and the other one is its dependent.

Surface syntactic analysis is done with UDPipe (Straka and Straková, 2017)[22]. UDPipe is language-independent and completely trainable for any language on annotated data in the CoNLL-U format[23]. Pretrained models are available for almost all languages included in the Universal Dependencies[24] (UD).

Figure 1 shows the output, in the form of a dependency tree, produced by UDPipe for the sentence "*Le strade attorno alla casa sono completamente allagate*" (The streets around the house are completely flooded).

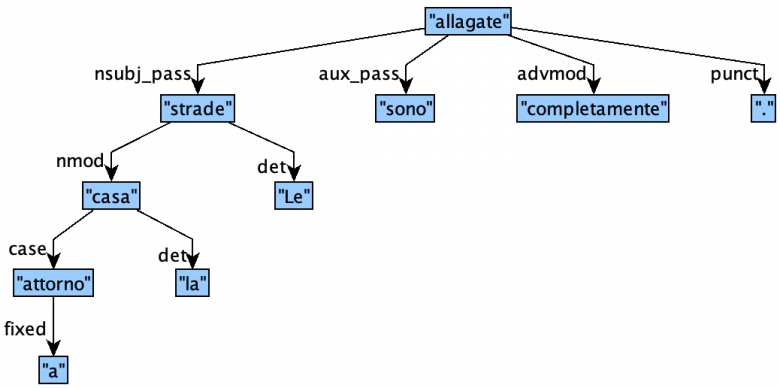

Figure 1: The dependency tree for the sentence "*Le strade attorno alla casa sono completamente allagate*".

3.7 Semantic Parsing

Finally, semantic analysis generates structured representations that will be stored in the Knowledge Base. For this task a set of graph-transduction grammars perform semantic parsing on top of SSynt representations produced by UDPipe. The pipeline outputs the semantic structures at two different levels of representation: deep-syntactic (or shallow-semantic) structures and semantic structures. An additional module extracts the targeted relations.

Deep-syntactic (DSynt) structures, produced on top of the SSynt representations, are language-independent syntactic trees with coarse-grained relations over the content words of a sentence (i.e., verbs, nouns, adjectives, adverbs). In DSynt structures, grammatical words (i.e., functional prepositions, conjunctions, auxiliaries, and determiners) are removed, all nodes have a Part-of-Speech tag, and dependency labels are oriented towards predicate-

[22] https://github.com/ufal/udpipe
[23] http://universaldependencies.org/format.html
[24] https://universaldependencies.org/

argument relations, and they include: I, II, III, IV, V, VI, for the arguments of a predicate; ATTR and APPEND, for adjuncts; COORD, for coordination; and NAME, for proper names.

Figure 2 shows the DSynt structure of the sentence "*Le strade attorno alla casa sono completamente allagate*" that corresponds to the SSynt structure shown in Figure 1 produced by the transduction grammars. Here, determiners and auxiliaries are removed, the subject is correctly identified as the second argument of the passive verb, and adjuncts are linked to their modified element with the non-core ATTR relation.

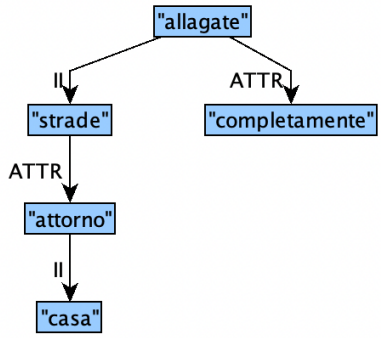

Figure 2: Deep-syntactic structure of the sentence "*Le strade attorno alla casa sono completamente allagate*"

Our graph-transduction grammars, responsible for the mapping process, are rules that apply to a subgraph of the input structure and produce a part of the output structure, mapping edges, adding attribute/value pairs, and removing nodes. Figure 3 is a sample rule from the SSynt-DSynt mapping. This rule replaces definite articles with an attribute-value pair. On the left-hand side, it matches a node *?Xl* that has a dependency *det* to the Italian article 'il' (specified in the *conditions* field). On the right-hand side, the dependency *?Yl* is removed and the new feature Definite = "Def" is added to the nouns. As a result of the application of this rule, the article '*le*' is removed in Figure 2.

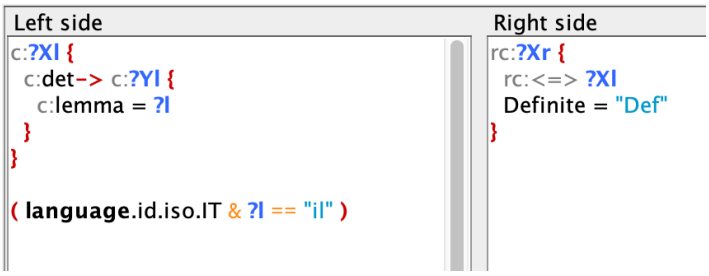

Figure 3: A sample graph-transduction rule; ? indicates a variable; ?Xl{} is a node, ?r-> is a relation, a=?b is an attribute/value pair.

Semantic structures are directed acyclic graphs with predicate-argument relations over the content words of a sentence. They are obtained by another sequence of graph-transducers

that apply on the deep-syntactic structures. Here, the deep syntactic core relations *I, II,… VI* are mapped to *Argument1, Argument2... Argument6*; the non-core relation *ATTR* is inverted and relabelled as *Argument1* when the governor is an argument of the dependent, or *nonCore*, otherwise; the lexical relation *NAME* connecting parts of proper nouns is maintained; finally, coordinating conjunctions are represented as predicates that have all conjuncts as arguments, labelled as *Set*.

Figure 4 shows the semantic representation of "*Le strade attorno alla casa sono completamente allagate*" obtained from the deep syntactic structure shown in Figure 2.

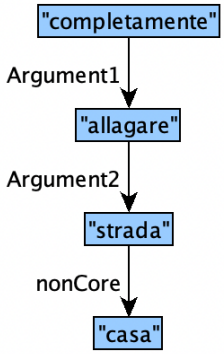

Figure 4: Semantic structure of the sentence "*Le strade attorno alla casa sono completamente allagate*"

The last step in language analysis is the mapping of the semantic structures in the form of predicate-argument structures onto **entity-relation-entity triples**. The main advantage of using this output representation is to reduce the gap between linguistic structures as provided by language analysis tools and the Knowledge Base representations.

For xR4DRAMA we target some specific features that allow to categorize the reports from citizens according to the situation reported and to identify the elements at risk and their location for PUC1. For PUC2, we are interested in those features related to the availability and accessibility of the specific facilities users are interested in, e.g., power outlets, bathrooms, restaurants, and cafés, etc.

This task is done by an additional component in the graph transducer that identifies the patterns in the predicate-argument structures that correspond to the targeted information and translates the entities through a simple dictionary lookup.

Figure 5 shows the triple extracted from semantic structure shown in Figure 4.

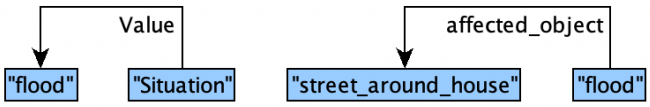

Figure 5: Entity-relation-entity triple extracted from the semantic structure shown in Figure 4

Table 6 sums up the current state of the graph-transduction grammars and rules for the mapping between SSynt structures and entity-relation-entity triples through DSynt and semantic structures we have implemented in the first period of the project for two of the languages of the project: English and Italian.

Table 6: Graph-transduction rules mapping. *Includes rules that simply copy node features (about 40 per grammar).

Grammar	#rules*	Description
0-Ud_normalization.rl	81	Normalise UD structures
1-UD_Track2_preproc.rl	103	Identify nodes to be removed Identify verbal finiteness and tense
2-UD_Track2.rl	159	Remove idiosyncratic nodes Establish correspondences with surface nodes Replace determiners, modality, aspect, and voice markers by attribute-value features Identify duplicated core dependency labels below one predicate
3-UD_postproc.rl	94	Replace duplicated argument relations Identify remaining duplicated core dependency labels
4-UD2MTT.rl	122	Assign DSynt dependencies Identify conjunct nodes
5-DSynt-Sem.rl	95	Recover shared arguments Establish coord. conjuncts as predicates
6-Sem-postproc.rl	78	Assign PredArg arguments
8-xr4drama-triples.rl	37	Map PredArg structures onto entity-relation-entity triples

4 CONCLUSIONS

In this deliverable we presented the initial version of the xR4DRAMA language technology modules for Automatic Speech Recognition (Section 2) and Language Analysis (Section 3), which covers: concept extraction (3.1) and named entity recognition (3.2), to detect and classify linguistic expressions that indicate relevant entities; temporal expression identification (3.3), to extract and normalize temporal expressions; word sense disambiguation (3.4) and geolocation (3.5), to annotate input texts with references to lexical meanings in databases, while also providing the respective geographic coordinates; surface language analysis (3.6), for the linguistic analysis from tokenization up to surface syntax (dependency) parsing; and deep analysis (3.7), to generate structured representations that will be stored in the Knowledge Base.

5 REFERENCES

Béchet, F., and Raymond, C. (2019). "Benchmarking benchmarks: introducing new automatic indicators for benchmarking Spoken Language Understanding corpora". In Proceedings of INTERSPEECH 2019, Graz, Austria.

Biadsy, F., Weiss, R. J., Moreno, P. J., Kanvesky, D., and Jia, Y. (2019). "Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-Impaired Speech and Speech Separation". arXiv preprint arXiv:1904.04169.

Boersma, P. & Weenink, D.. (2021). Praat: doing phonetics by computer [Computer program]. Version 6.1.54, retrieved 9 October 2021 from http://www.praat.org/

Camacho-Collados, J., Taher Pilehvar, M. and Navigli, R., Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artificial Intelligence 240, Elsevier, 2016, pp.567-577.

Casamayor, G., Semantically-oriented text planning for automatic summarization (Doctoral dissertation, Universitat Pompeu Fabra), 2021.

Godfrey, J., Holliman, E., and McDaniel, J. (1992). "SWITCHBOARD: telephone speech corpus for research and development". In Proceedings of the IEEE international conference on Acoustics, speech, and signal processing (ICASSP'92), Vol. 1. IEEE Computer Society, Washington, DC, USA, pp.517-520.

Gu, J., Z. Lu, H. Li and V.O.K. Li. (2016). "Incorporating copying mechanism in sequence-to-sequence learning". *Proceedings of the ACL*.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates,A., Ng, A.Y. (2014). "Deep Speech: Scaling up end-to-end speech recognition". In: https://arxiv.org/pdf/1412.5567v2.pdf

Iancu, B. (2019). "Evaluating Google Speech-to-Text API's Performance for Romanian e-Learning Resources," Informatica Economica, Academy of Economic Studies - Bucharest, Romania, vol. 23(1), pages 17-25.

Navigli, R. and Ponzetto, S.P., 2010, July. BabelNet: Building a very large multilingual semantic network. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 216-225).

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., Collobert, R. (2019). "Wav2Letter++: A Fast Open-source Speech Recognition System," In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 6460-6464. DOI: 10.1109/ICASSP.2019.8683535.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., Collobert, R. (2020). "MLS: A Large-Scale Multilingual Dataset for Speech Research". In Proceedings of Interspeech 2020, 2757-2761. DOI: 10.21437/Interspeech.2020-2826.

Rousseau, A., Deléglise, P., and Estève., Y. (2012). "Tedlium: an automatic speech recognition dedicated corpus". In Proceedings of LREC 2012, pp. 125-129.

See, A., P.J. Liu, and C.D. Manning. Get to the point: Summarization with pointer-generator networks. Proceedings of the ACL, 2017: 1073-1083.

Straka, M. and Straková J. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 2017.

Strötgen, G.: HeidelTime: High Qualitiy Rule-based Extraction and Normalization of Temporal Expressions. SemEval'10.

Vasilescu, I., Yahia, D., Snoeren, N., Adda-Decker, M., and Lamel, L. (2011): "Cross-lingual study of ASR errors: on the role of the context in human perception of near-homophones", In Proceedings of Interspeech 2011, pp. 1949-1952. Weischedel, R., et al. OntoNotes Release 5.0 LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium, 2013.

Weischedel, R., et al. OntoNotes Release 5.0 LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium, 2013.

Yashesh Gaur, W., Lasecki,S., Metze, F., and Bigham, J.P. (2016). "The effects of automatic speech recognition quality on human transcription latency". In Proceedings of the 13th Web for All Conference (W4A '16). ACM, New York, NY, USA, Article 23, 8 pages.