xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.12

Multilingual information generation techniques v2

Dissemination level:	Public
Contractual date of delivery:	Month 25, 30 November, 2022
Actual date of delivery:	Month 26, 7 December, 2022
Work package:	WP3 Analysis and fusion of multi-modal data
Task:	T3.6 Personalised information generation
Type:	Demonstrator
Approval Status:	Final version
Version:	1.0
Number of pages:	56
Filename:	D3.12_xR4Drama_MultilingualInformation GenerationTechniquesV2_v1.0.docx

Abstract

Deliverable 3.12 describes the advanced versions and outcomes of the natural language generation components of xR4DRAMA developed in task T3.6 “Personalised information generation” of WP3. This task accounts for the production of multilingual text generation in the xR4DRAMA platform. The component involved in this task is the report/text generation module, which receives input from the multimodal information fusion (T3.5).

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	28/10/22	Table of contents.	Alba Táboas
0.2	09/11/22	Added complete Section 2 on the description of tools, introduction to section 4 on improvements, and introduction to section 5 on neural experiments.	Alba Táboas
0.3	16/11/22	Added Section 3 on content selection with output examples, and Section 5 on neural experiments and their application to xR4DRAMA .	Alba Táboas
0.4	23/11/22	Finished section 4 on system improvements and added sections 1, 5 and 6, plus references.	Alba Táboas
0.5	25/11/22	Section on integration in xR4DRAMA platform and workflow. Draft for internal review by CERTH.	Jens Grivolla
0.6	01/12/22	Revision based on review comments.	Alba Táboas
1.0	07/12/22	Final formatting / metadata.	Jens Grivolla

Author list

Organisation	Name	Contact Information
UPF	Alba Táboas	alba.taboas@upf.edu
UPF	Jens Grivolla	jens.grivolla@upf.edu

Executive Summary

Deliverable 3.12 describes the progress on the task T3.6 “Personalised information generation” of WP3. This task accounts for the production of multilingual text generation in the xR4DRAMA platform.

The component involved in this task is the report/text generation module, which receives input from the multimodal information fusion (T3.5) to be realised as natural language sentences. The advances during the second half of the project are discussed in the course of the deliverable, namely:

(i) the definition of the selected information to be covered by the xR4DRAMA platform on the generation side for both use cases,

(ii) the extension and improvement of the linguistic resources for natural language generation (conceptual templates, lexica, graph transduction grammars) for English and Italian, and

(iii) some experiments on a hybrid approach combining grammar-based generation and neural paraphrasing techniques.

Abbreviations and Acronyms

DB	DataBase
FORGe	Fabra Open-source Rule-based Generator
GIS	Geographic Information System
JSON	JavaScript Object Notation
KB	Knowledge Base
LAS	Language Analysis System
MT	Machine Translation
NLG	Natural Language Generation
NMT	Neural Machine Translation
OSM	OpenStreetMaps
POI	Point of Interest
PoS	Part of Speech
PUC	Pilot Use Case
VA	Visual Analysis
WP3	Work Package 3

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

In this deliverable, we report on the work carried out to improve xR4DRAMA's multilingual information generation, which corresponds to task 3.6 in WP3, mostly during the second half of the project's development. The final goal was to develop a Natural Language Generation system that satisfies the users' needs regarding the verbalisation of information in the shape of multilingual reports. UPF's generator takes as input a selection of contents from the Knowledge Base (KB, T3.5) and produces texts in English and Italian.

Following the methodology previously described in the DoA and D3.6, our approach to text generation divides the task into three subtasks: content selection, discourse planning and linguistic generation. At this second stage in the project, a big part of our efforts was focused on content selection, i.e. on the decision of what information coming from the KB needs to appear verbally in the generated reports, a task which was addressed in accordance with user needs for both pilot use cases. Discourse planning (also referred to as "sentence packaging", "text planning" or simply "aggregation") and linguistic generation were tackled by improving UPF's grammar-based generator FORGe (Mille et al., 2019). The improvement consisted in increasing its coverage both for English and Italian (although a large part of the rule engine is multilingual) by extending our linguistic resources and by experimenting with the implementation of a complementary module using neural networks.

The remainder of this document is organised as follows: first, we present an overview of the approach adopted for text generation and its implementation (Section 2); then, a description of the two use cases involved in the project, their generation needs and examples of generated reports (Section 3), followed by a summary of the improvements made to the grammar-based generator (Section 4); after that, we provide a brief report on the results obtained from experiments on a hybrid approach combining the rule-based generator with state-of-the-art machine learning techniques (Section 5); and finally we discuss the results obtained from a preliminary evaluation process (Section 6).

2 BRIEF DESCRIPTION OF TEXT GENERATION TOOLS

This section provides a summary of how the text generation pipeline is structured and a brief description of the tools that constitute it.

The traditional view of Natural Language Generation (NLG) addresses this complex and challenging task as a sequence of three subtasks: (i) content selection, which is responsible for deciding the specific subset of available contents to be conveyed in the generated text, (ii) text planning, which takes care of packaging these contents into discursively organised units (i.e., sentences), and (iii) linguistic generation, which provides the surface realisation of the contents as well-formed text (Rambow and Korelsky, 1992).

The approach we adopted for the development of the NLG pipeline for xR4DRAMA follows this tradition. The pipeline consists of two main components: a knowledge-oriented module or interface with the project's Knowledge Base (KB), which is in charge of subtask (i), and a linguistics-oriented module, the text generation module, responsible for subtasks (ii) and (iii).

The knowledge-oriented module selects the relevant contents from the KB (and only those) according to users' requirements for each use case and its sub-cases, and then stores them in a structured JSON file which serves as input to the text generation module. The text generation module takes this input and performs a series of transformations, organising the selected contents into well-formed sentences that compose the resulting text.

This second module is quite complex and constitutes the system's main text generation tool. The rest of the section is devoted to its description, but for further information and a more detailed explanation, with concrete examples, see D3.6.

2.1 Grammar-based generation

2.1.1 Approach

Text generation, as we approach it, is divided into two sub-modules: one is in charge of sentence packaging (or text planning) and the other is in charge of linguistic generation, which, in turn, is split into several components tackling the tasks of sentence structuring (i.e. lexical selection and syntactic organisation), word ordering and morphological agreement resolution. This division of text generation into specific tasks conforms to the ideas of the Meaning-Text Theory (Mel'čuk, 1988), which advocates for a precise and independent modelling of each level of language description (semantics, syntax, topology, morphology). The generation is performed on successive steps mapping one level of representation onto the next and the two submodules work together.

First, the **ontology constructs** contained in the input coming from the KB are mapped onto **conceptual structures** of a linguistic nature. All the elements involved are mapped to predicates or arguments (i.e., nodes and/or labelled edges that link the predicate to the argument). The results are simple predicate-argument templates associated with each data point in the KB.

These conceptual structures are then mapped to **semantic structures**, language-specific structures in which the basic meaning units (semantemes) are lexical units in the target language. Semantic structures are unambiguous, each semanteme being the argument of a predicate and numbered by its valency through the relation linking them both (subcategorization frame). All the nodes are assigned a PoS tag and a specific entry in a lexical resource.

At this point, the text planning sub-module comes into play by grouping together some of the resulting semantic structures and determining the boundaries of the sentences. Whenever possible, different discursive units are combined or “aggregated” to form complex sentences. This is done in two steps: first, looking for shared elements, same predicate and subject/object or location, to aggregate through coordination, and then, checking if an argument is repeated in different discursive units so that subject/object progression can be performed. This is also the stage at which the communicative structure is determined, identifying theme, rheme and eventual specifiers in each sentence.

The resulting packaged semantic structures are mapped onto **deep-syntactic structures**, step at which the sentence structure is defined. The semantic graph is transformed into a tree whose root is the main part of the rheme and that is built node by node starting from its root. At this stage, the aforementioned lexical resource (or lexicon) is used to determine the syntactic predicate requirements, support verbs are introduced, and co-referring nodes are linked together.

Afterwards, deep-syntactic structures are mapped to **surface-syntactic structures** containing all idiosyncratic information. At the previous stage only meaningful lexical units and abstract lexemes (formalised as lexical functions) were present, now it is the moment in which non-meaningful units are introduced according to the information in the lexicon: governed prepositions, conjunctions, determiners, auxiliary verbs, expletive subjects, etc. Lexical functions are realised as the appropriate lexical units here. Generic syntactic relations (argument numbers) are also transformed into more specific relations conveying accurate syntactic information (subject, object, noun modifier, etc.). Feasible pronominalisations and elisions are performed at this level to avoid repetitiveness.

Sentence packaging is activated again at this moment, so that aggregations that could not take place in the deeper levels because the elements had not been introduced yet (such as expletive subjects, support verbs or lexical functions) can be performed now.

From here, word ordering and word agreement are resolved to obtain **morphologic structures**. The morphological information of each word is gathered at this stage, which is done using a hybrid method: regular inflection rules are applied, when possible, otherwise specific entries are looked up in a morphological dictionary. Finally, punctuation marks are introduced, and the final form of the words is retrieved to provide the outcome **sentence(s)**.

2.1.2 Implementation

The succession of mappings that constitute xR4DRAMA’s text generation module is rendered by a grammar-based system called FORGe (Mille et al., 2019). This system is an open-source generator developed by UPF and implemented as a graph-transducer platform, BUDDY, which is, in turn, a faster and more efficient re-implementation of the MATE platform (Bohnet and Wanner, 2010).

The basic text generation implementation consists of manually crafted graph-transduction grammars for each transition between two consecutive layers. In combination with the rules, dictionaries of two different types are required: one that describes the syntactic properties of these words (lexicon), and one that contains the inflection patterns of each word (morphologicon).

Improvements to this system entail the manufacture of conceptual structures, the creation of new rules and modification of old ones to extend the coverage of these grammars, as well as the expansion of the morphological and lexical dictionaries with the addition of new entries. The specific improvements carried out during this project are detailed in Section 4.

2.2 Integration in the xR4DRAMA platform and workflow

The text analysis services, both for speech recognition and text analysis are packaged and deployed as Docker[1] containers, running on Docker Swarm[2]. They are accessible as REST-like[3] web services, with Swagger[4]-based documentation and an interactive web-based test interface available at **https://xr4drama.upf.edu/xr4drama-services/.**

These services are called as needed by other components of the xR4DRAMA platform, receiving a JSON structure with the input data and returning a JSON structure that contains the generated text, to be then stored in the platform or displayed to the user.

As described in more detail in section 3, NLG is applied in four different ways, two for each use case. For each use case, NLG is used on one hand for generating reports, giving the user an overview of a situation, and on the other hand for generating the descriptions used for creating points-of-interest (POIs) in the platform: geolocated specific information that can be visualised on a map through the different interfaces of the xR4DRAMA platform.

[1] https://www.docker.com/

[2] https://docs.docker.com/engine/swarm/

[3] https://en.wikipedia.org/wiki/Representational_state_transfer

[4] https://swagger.io/

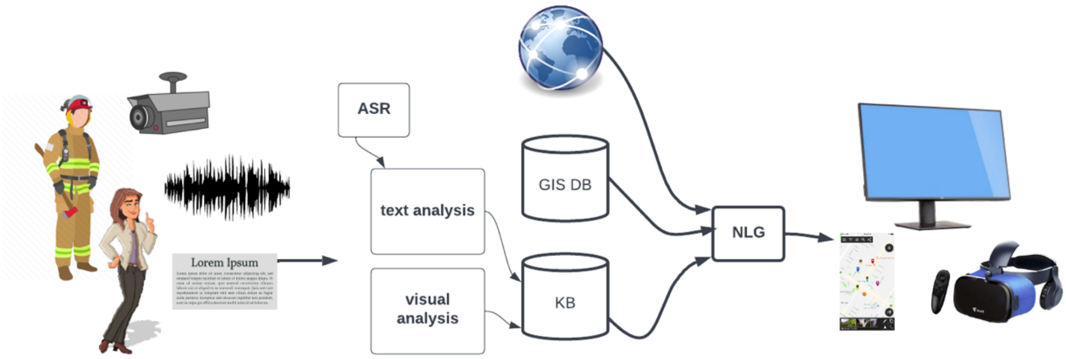


Figure 1: Information flow in the xR4DRAMA platform

Information from different sources, including texts and images analysed by the corresponding analysis services, as well as geographical information from sources such as OpenStreetMaps[5] (OSM) is accumulated in the project databases, in particular the GIS DB and the KB. NLG is then used to make that aggregated information accessible to end users in readable form.

POIs are created whenever relevant geolocated information enters the system. For PUC1 that consists particularly in analysis results based on images or text/audio messages from citizens, first responders, or other cameras. For PUC2 on the other hand, Foursquare[6] and similar sources can provide information on the availability of facilities of interest for media production. Upon receiving this information (and possibly combining it into a single POI), the KB calls the NLG service to create a readable brief description that is then used to create the corresponding POI in the system.

Additionally, two types of reports can be generated. For PUC2, a report with relevant information about a media production location is generated using information from the KB, GIS service as well as external services on the internet. For this, the NLG service is called (on user demand) with the corresponding project ID (which defines the location and gives access to previously acquired data), the required data is retrieved, and a generated report returned to be presented to the user. For PUC1, again on user demand, the NLG service is called with the project ID, start and end timestamps. The corresponding information (in particular text analysis and visual analysis results) is then retrieved, and the NLG service returns a report presenting a timeline of events for that timeframe and location, for the user that requested the report.

[5] https://www.openstreetmap.org/#map=7/38.359/23.810
[6] https://foursquare.com/

3 GENERATED CONTENTS

In this section, we provide a brief description of the two use cases involved in the project with their different text generation needs, we list the selected contents to appear in the generated texts in each case and we show some examples of the resulting texts.

3.1 Pilot Use Case 1 - Disaster management

The first pilot use case (from here on, PUC1) concerns the management of flood emergencies in a highly populated and urbanised area. The aim of xR4DRAMA's platform in this use case is to improve situation awareness during the disaster management process. This is done by supporting and facilitating, through the use of several techniques (including extended reality technologies) the activities and interventions carried out both by managers in the control room and first responders in the field.

The platform includes a map showing information coming from several sources, such as visual analysis of videos or images captured by static cameras, sensor analysis from water level detectors, flow velocity and water depth raster data, audio and text analysis from messages sent by citizens, datasets including the location of cultural heritage or natural sites, among others. This information can be shown as affected regions in the map or as geo-located points of interest (POIs).

Regarding text generation, this use case has two different needs. First, information about POIs needs to be presented in an interpretable way for users. This need is addressed by generating a title and a description of the emergency situation that caused the creation of the POI. Second, users may be interested in having a summary of what has happened during a predetermined period of time. To satisfy this other need, a timeline report is created, conveying a slightly shorter description of the emergency situations that occurred during that period ordered by the time at which they happened.

3.1.1 Information sources

As mentioned above, there are many possible sources of information relevant to PUC1 within xR4DRAMA's platform. Not all this information and data will be displayed in text form, since it was decided to be better shown to the users in other formats (for instance, visually through the map).

All the information that is covered by the NLG component for this use case comes from visual analysis of images captured by cameras and text analysis of warning messages sent by citizens through a specially designed app.

A sample of an input json file for the text generation module can be found in Appendix A.1.

3.1.2 Covered data

Table 1 below contains an exhaustive list of the information or data points covered by xR4DRAMA's text generation component for the creation of POIs and their description, together with the text generated for each data point individually. Note that: (i) text between square brackets are place holders that are later substituted with the actual information values, and (ii) some data points are associated with more than one possible text, the choice between one or the other depends on the condition specified in the middle column, mainly, the availability, type or confidence level of the information being verbalised.

Data point	Condition	Generated text
POI title	One source of info (either VA or LAS)	[quantity] [subCategory] in danger.
	Several sources of info available (both VA and LAS)	Between [quantityMin] and [quantityMax] [subCategory]s in danger.
Emergency and area (as detected by VA)	¬ Emerg. type = none Conf. level > 0.8	The visual analysis has detected a [emergency] in a [area].
	¬ Emerg. type = none Conf. level < 0.8	The visual analysis seems to have detected a [emergency] in a [area].
	Emerg. type = none Conf. level > 0.8	The visual analysis has not detected any emergency.
	Emerg. type = none Conf. level < 0.8	The visual analysis doesn't seem to have detected any emergency.
Elements in danger (as detected by VA and LAS)	One source of info	[quantity] [subCategory]s are in danger.
	Several sources of info available	Between [quantityMin] and [quantityMax] [subCategory]s are in danger.
	Quantity = 0	No [subCategory]s are in danger.
River overtopping (as detected by VA)	Info available	The river has (not) overflowed.
Emergency situation reported (as detected by LAS)	Info available	A [situation_N] has been reported. *
Object(s) and agent(s) involved (as detected by LAS)	Only object(s) detected	[numberAffectedObject] [affectedObject]s are/have [situation_V].**
	Both object(s) and agent(s) detected	[agent] has [situation_V] [numberAffectedObject] [affectedObject]s.**
Situation location (as detected by LAS)	Info available	There is a [situation] in a [location].
Situation severity (as detected by LAS)	Info available	The [situation] is [severityDegree].

* [situation] is realised as a noun. Ex: *"A flood has been reported."*
** [situation] is realised as a verb in its participle form. Ex: *"Parco Querini is flooded." / "The water has dragged two cars."*

Table 1: List of covered data points - PUC1

To cover the other need regarding generation for PUC1, we need to create a text in the shape of a timeline report. In the report, these same data points are covered, but we obviously need to include time information as well. There are two small differences regarding the data points and the text generated for each of them: all sentences are generated in the past tense and only information about existence is included in the report, so, for instance, if no element (people, vehicles, infrastructures, etc,) is in danger the negative sentence does not appear in the report.

3.1.3 Sample generated texts

Although each data point has an individual verbal realisation (the rightmost column in Table 1, or a slight variation of this in the case of the timeline report), a list of such sentences would result in a repetitive and mechanical sounding text. The language generation module groups together some of the data points in longer and more natural sounding sentences to produce a cohesive and coherent text, thus presenting the information in a more user-friendly way.

Below there is an example of the text generated for a POI created from VA and LAS results, with its title and description, both in English and in Italian:

English text:

> *<u>2 people and 1 vehicle in danger</u>*
>
> *The visual analysis seems to have detected a flood at an urban canal and a flood has been reported. 3 people and 4 vehicles are in danger, but no animals are in danger. The river has overflowed. A parking lot is flooded in piazza Matteotti.*

Italian text:

> *<u>3 persone e 4 veicoli in pericolo</u>*
>
> *L'analisi visuale sembra avere rilevato un'alluvione in un canale e un'alluvione è stata segnalata. 3 persone e 4 veicoli sono in pericolo ma nessun animale è in pericolo. Il fiume non è esondato. Un parcheggio è allagato in piazza Matteotti.*

And now we present an example of a short timeline report corresponding to emergencies happening for one day. The report is available in English and in Italian.

English report:

> *<u>October 13, 2022</u>*
>
> *At 10:30 the visual analysis had not detected any emergency, but an obstruction had been reported. 2 people and 1 vehicle were in danger. A car and 2 people were blocked at a bridge.*
>
> *At 10:38 the visual analysis had detected a flood in a formal garden and a flood, and an entrapment had been reported. 10 people and 2 vehicles were in danger. Parco Querini was flooded. 10 people were trapped.*

At 11:00 the visual analysis seemed to have detected a flood at an urban canal and a flood had been reported. 3 people and 4 vehicles were in danger. The river had overflowed. A parking lot was flooded in piazza Matteotti.

Italian report:

13 Ottobre, 2022

Alle 10:30 l'analisi visuale non aveva rilevato nessuna emergenza ma un'ostruzione era stata segnalata. 2 persone e 1 veicolo erano in pericolo. Un'auto e 2 persone erano bloccate a un ponte.

Alle 10:38 l'analisi visuale aveva rilevato un'alluvione in un giardino all'italiana e un'alluvione e un intrappolamento erano stati segnalati. 10 persone e 2 veicoli erano in pericolo. Parco Querini era allagato. 10 persone erano intrappolate.

Alle 11:00 l'analisi visuale sembrava avere rilevato un'alluvione in un canale e un'alluvione era stata segnalata. 3 persone e 4 veicoli erano in pericolo. Il fiume era esondato. Un parcheggio era allagato in piazza Matteotti.

3.2 Pilot Use Case 2 - Media production

The second use case (or PUC2 from now on) concerns media production planning from a remote position. The goal of the project in this case is to support and facilitate the planning activities that need to be carried out by the production management team by improving their situational awareness despite their not being at the chosen location.

Similarly, to the other use case, PUC2 also makes use of a map, in this case of the shoot's intended location, showing relevant information for the production planning. POIs in this map are created from the information extracted via web crawling, visual and text analysis, and other sources. These POIs, like those in PUC1, need a title and a description, for which the text generation module is used.

The NLG module is used as well to satisfy another important generation need for media production planning: the creation of a preliminary report containing general information about the selected location, weather forecasts, as well as practical information regarding the availability of facilities in the area (such as the presence of hospitals, fire departments, accommodation options, etc.).

3.2.1 Information sources

The information that is covered by the NLG component for this use case comes mainly from two different sources: data available in xR4DRAMA's KB extracted through web crawling of sites like Wikipedia[7] or OpenStreetMap[8], as well as text analysis of comment messages sent by citizens through FourSquare[9] and similar platforms.

[7] https://en.wikipedia.org/wiki/Main_Page
[8] https://www.openstreetmap.org/#map=7/38.359/23.810
[9] https://foursquare.com/

A sample of an input json file for the text generation module can be found in Appendix A.2.

3.2.2 Covered data

All the data points covered by our NLG module for this use case, together with the text generated for each of them individually, are listed in the following table. Recall that words between square brackets are place holders that are later substituted with the actual information values, and that the condition in the middle column determines what of the two verbalisation options will be used.

Data point	Condition	Generated text
Quantity and Type of POIs for Preliminary Report	numberSubCategory > 0	There are [numberSubCategory] [subCategory]s in the selected area.
	numberSubCategory = 0	There are no [subCategory]s in the selected area.
Availability Info for POI title		Access to [subCategory]
Availability Info for POI description	class info available	[subCategory]s are available in this [subCategoryClass].
	no class info	There is access to [subCategory]s here.
SubCategory Quality		The [subCategory] is [quality].
SubCategory Quantity	class info available	There are [numberSubCategory] [subCategory]s in this [subCategoryClass].
	no class info	There are [numberSubCategory] [subCategory]s here.
SubCategory Location	only location no relative loc info	The [subCategory]s are located at a [subCategoryLocation].
River overtopping (as detected by VA)	there is a relative position extracted	The [subCategory]s are located at a [rel] [subCategoryLocation].

Table 2: List of covered data points - PUC2

Note that only the first row in the table corresponds to information that will be part of the preliminary report containing general information about the selected location. However, this is not the only information that the report includes. In fact, it is made up of three sections: first, the introductory paragraph of the location's Wikipedia page (with a direct link to the source material); then, a brief summary with weather information in textual or visual form (also including the source's link); and, finally, a list of sentences regarding the quantity and type of POIs already stored in the platform server, which are generated using our NLG module, and correspond to the data points in the first row of Table 2.

The rest of the data points in Table 2 are used for the creation of POIs, with their title and description.

3.2.3 Sample generated text

The figure below shows an example of a preliminary report for media production planning on the city of Corfu in the Ionian Islands (Greece).

Media Production Report - Corfu Town

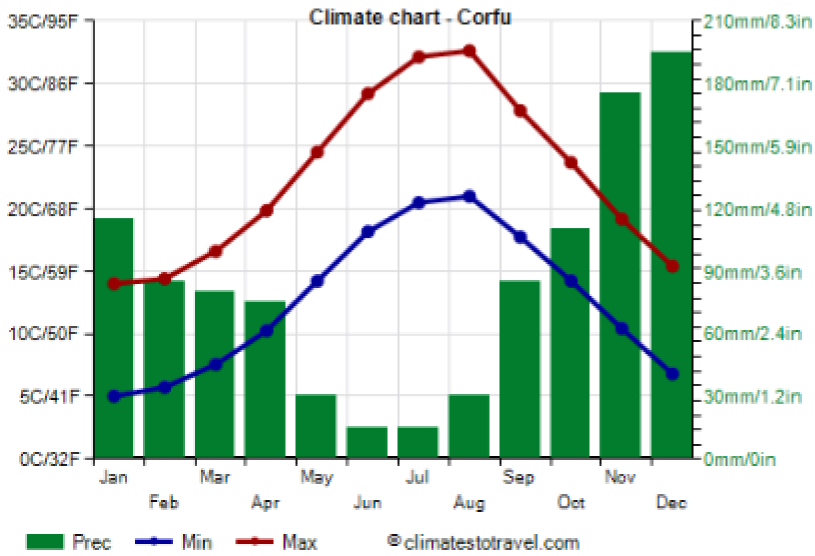
General Information

Corfu is a city and a former municipality on the island of Corfu, Ionian Islands, Greece. Since the 2019 local government reform, it is part of the municipality of Central Corfu and Diapontia Islands. It is the capital of the municipality and of the Corfu regional unit. The city also serves as a capital for the region of the Ionian Islands. The city (population in 2011: 39,674 residents and the whole island 111,975) is a major tourist attraction and Greek regional center and has played an important role in Greek history since antiquity.

Source: https://en.wikipedia.org/wiki/Corfu_(city)

Weather

The climate of Corfu (or Kerkyra), a Greek island in the Ionian Sea, is Mediterranean, with mild, rainy winters and hot, sunny summers. The daily average temperature ranges from 9.5 °C (49 °F) in January to 27 °C (80.5 °F) in August.



Source: https://www.climatestotravel.com/climate/greece/corfu

Figure 2: Sample Media Production Preliminary Report

Practical Information

Food and drink
There are 35 cafés, 43 restaurants and 55 bars and pubs in the selected area.
Accommodation
There are 5 hotels but no apartments and no campsites in the selected area.
Commerce
There are 21 grocery stores, 23 clothing stores, 2 electronics stores and 1 other shop in the selected area.
Finance
There are 2 ATMs and 12 banks but no exchange offices in the selected area.
Facilities
There is 1 toilet, 35 drinking water access points, 1 gas station and 1 charging station but no plugs and no internet access points in the selected area.
Public services
There are no tourist offices, no fire departments and no cemeteries but 1 police station and 17 public buildings in the selected area.
Medical
There are 22 pharmacies and 2 hospitals but no doctor's offices in the selected area.
Transportation
There are 13 parking lots, 93 public transport points and 23 airports but no taxi stands, no car rentals and no bike rentals in the selected area.
Cultural
There are 69 sights, 7 museums and 8 venues in the selected area.

Source: xR4DRAMA's DB

Figure 3: Sample Media Production Preliminary Report (continuation)

Now we present two examples of the information in POIs concerning PUC2, with their title and description:

Access to Wi-Fi and plugs

Wi-Fi and some plugs are available in this café. The free Wi-Fi is unreliable, and it requires a password. The charging plugs are located under the table.

Access to parking

There is access to valet parking here. The parking is free. It is located nearby behind the hotel.

4 IMPROVEMENTS TO GRAMMAR-BASED GENERATION

During the first 12 months of development, the main improvements made to UPF's FORGe multilingual discourse generator both to address xR4DRAMA's project-specific requirements and to enhance the system's general quality and portability can be summarised as follows:

- initial predicate/argument templates and lexical resources had been crafted;
- rules were extended to cover project-specific types of sentence packaging so as to make the generated texts more fluent; the general coverage of the rules had been improved (the number of rules increasing from 1,995 to 2,147);
- a new morphology generation module had been implemented;
- rules were generalised, making them more language independent: 74% of the rules were language-independent, as opposed to 70% at the beginning of the project;
- updated grammars resulted in a significant improvement of the generator's quality on a challenging benchmark dataset for English: WebNLG+2020 (Castro Ferreira et al., 2020);

In addition, new datasets for training and testing statistical generation tools had been developed and released publicly. Methodologies for evaluating the outputs of generation systems had been proposed and validated in the context of a couple of international shared tasks. See D5.6 for an in-depth description of these improvements.

In this section, we describe the advances carried out during the second half of the project's development, providing details about the creation of manually crafted conceptual structures and the extension of lexical resources to fit the needs of both PUCs in xR4DRAMA, and also describing the new rules introduced to the generator's grammars especially for his project as well as those with a more generic scope, paying particular attention to the rules created to improve the generator's coverage for the English and Italian languages.

4.1 Conceptual structures

In order to generate readable texts containing the relevant information chosen by users, first the selected contents need to be converted into linguistically motivated structures; this is what we refer to as "conceptual" structures (introduced in D3.6 but see also Section 2.1).

The conceptual structures act as an interface between the KB and the text generation module. The name "conceptual" comes from the fact that they are language-independent and domain-agnostic: regardless of the target language or field of knowledge, the same vocabulary is used for the concepts and the labels of the relations linking them.

These structures are in the form of simple predicate-argument templates associated with one or a couple of individual data points from the KB, and each of them will be realised as an independent sentence (as seen on the right columns of Tables 1 and 2) or aggregated together into more complex sentences (as seen in the sample texts).

Figure 4 shows a conceptual (or predicate-argument) template for the type of emergency occurring at some area and one for the quantity of people/vehicles/animals/infrastructures currently in danger.

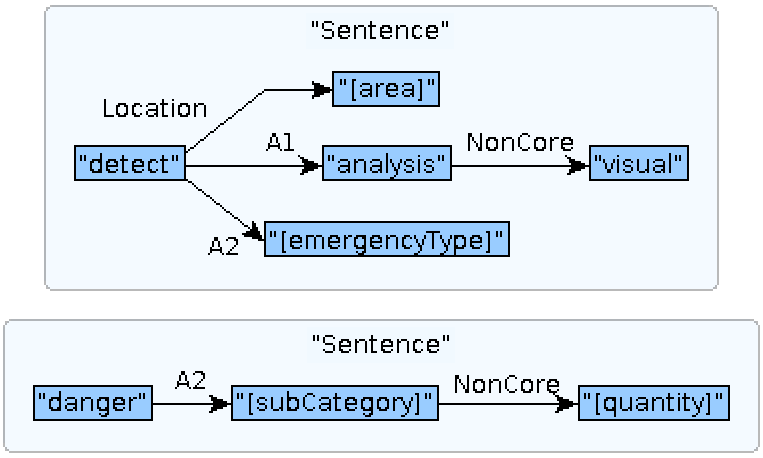


Figure 4: Sample conceptual templates

The templates are then populated with the information from the KB found in an input JSON file. For instance, the previous two templates can be instantiated as the three conceptual structures shown below by filling the variable slots with actual data.

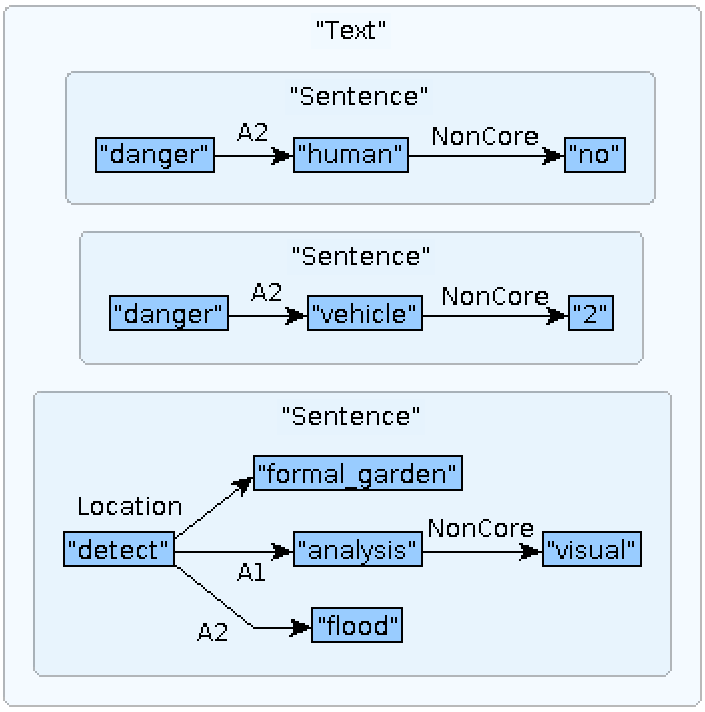


Figure 5: Sample conceptual structures

During the first development stage, 23 initial predicate-argument templates had been crafted to cover (most of) the data points required by the users (as found in D6.2). Taking into account the project's evolution during the second stage of development and considering how the user's needs changed regarding which information would be best conveyed as text, we built on the initial templates and crafted 42 additional ones, creating a total of 65, out of which, 37 ended up being used.

Most of the “discarded” templates were not used because they were substituted for more generic versions or the phrasing was slightly modified (Figure 6 shows an example of both), or because the information to populate them was not available at this point (Figure 7). In the latter case, the templates might still be used in the future iterations of the project.

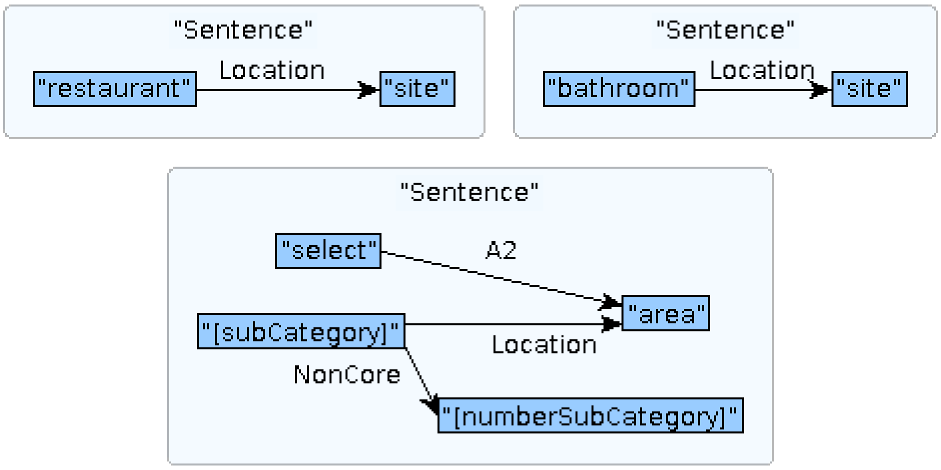


Figure 6: Sample of templates discarded in favour of a more generic one
There are restaurants on the site. > There are 12 restaurants in the selected area.

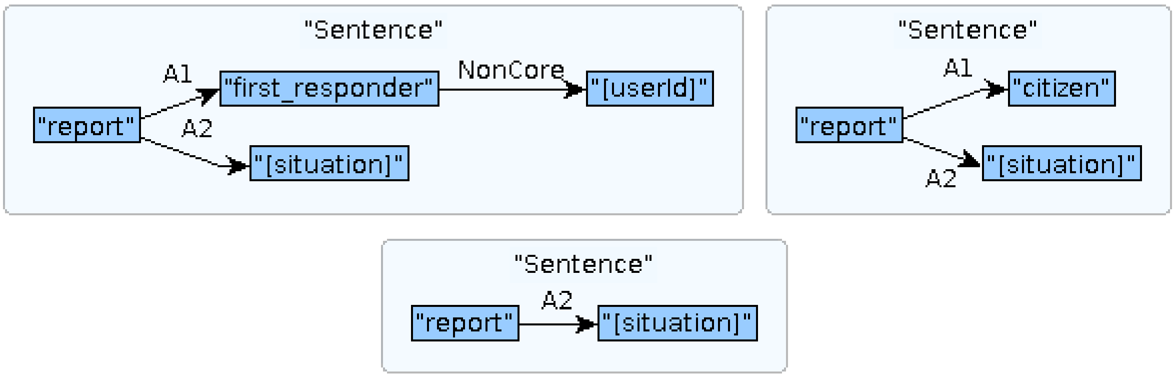


Figure 7: Sample of templates discarded in favour of a simpler one
First responder #7 has reported a flood. / A citizen has reported a flood. > A flood has been reported.

4.2 Extension of lexical resources

Recall that within our text generation module, conceptual structures are successively mapped onto structures belonging to other levels of linguistic representation, mainly, semantic, syntactic, and morphological (see section Section 2.1 for a brief explanation or D3.6 for a deeper understanding). These transformations require the use of good quality lexical resources or dictionaries.

More specifically, the module needs three types of dictionaries: (i) one that maps the input concepts onto lexical units of specific languages, which is called concepticon; (ii) several lexicons or monolingual dictionaries that contain the combinatorial properties of each lexical unit and linearisation properties of dependencies; and finally, (iii) monolingual dictionaries with the full forms of the words and/or specific morphemes, i.e. containing the relevant morphological information, which are called morphologicons. The concepticon is used to go from conceptual structures to semantic ones, lexicons are used in the mapping from semantic structures to syntactic ones, and morphologicons help transform morphologic structures into the text's final surface realisation.

Advances regarding these resources had already started in the first half of the project and have been resumed in this second half by manually crafting new entries for all three types of dictionaries widely covering the xR4DRAMA domain.

The concepticon is a multilingual dictionary in which the keys are predicates from the conceptual structures that serve as input to the text generation module, and the values are lexical units in other languages (note that several different lexical units can appear for the same language). On month 12, this dictionary contained 405 entries and it has currently been extended to 921 entries. Figure 8 shows an example of a concepticon entry.

```
"bridge" {
  EN = {
    lex = "bridge_NN_01"
  }
  IT = {
    lex = "ponte_NN_01"
  }
  EL = {
    lex = "γέφυρα_NN_01"
  }
  ES = {
    lex = "puente_NN_01"
  }
}
```

Figure 8: Conceptual entry "bridge" in the concepticon dictionary

Each lexical unit contained in the concepticon is a key in the corresponding lexicon, where these lexical units are described. The English lexicon had already been created from linguistic resources such as VerbNet, NomBank and PropBank with a hybrid approach involving both automatic conversion of the information in these resources as well as manually crafting entries. However, it has also been extended by adding entries needed to cover specific domains (including xR4DRAMA's domain). The Italian lexicon was manually crafted for this project. On month 12, the English lexicon contained 654 entries and it now contains 903, whereas the Italian lexicon contained 162 entries and it has been extended to 392. The two entries in Figure 9 below are an example of added entries to the English dictionary.

```
"flood_VB_01":_verbExtArg_ {
    vncls = "45.4"
    vncls = "9.8"
    pbID = "flood.01"
    pbsenseID = "01"
    lemma = "flood"
    gp = {
      I = {}
      II = {}
      III = {
        prep = "with"
        prep = "by"
      }
    }
  }
```

```
"flood_NN_01":_noun_ {
    vncls = "47.2"
    pbID = "flood.01"
    pbsenseID = "01"
    lemma = "flood"
    gp = {
      I = {
        prep = "of"
      }
      II = {}
      III = {}
    }
  }
```

Figure 9: New entries in the English lexicon to cover xR4DRAMA's needs

Finally, morphologicons contain the surface forms of inflected words, or morphemes when regular morphology can be applied. At the end of the first 12 months of the project, the English morphologicon contained 747 entries. This dictionary has been modified to include regular morphology and now, the number of entries has been reduced to 673.

It is important to mention that the number of entries in this kind of dictionaries is not representative of the improvements, since some of the original entries can be removed given that they are now covered by regular morphology rules and the morpheme information already suffices. In fact, this dictionary still has room for improvement by removing more of the entries that are no longer needed. The following figure shows a couple of entries with irregular morphology.

```
"give<V><IND><PAST><PL><3>" { word = "gave" }
"give<V><IND><PAST><SG><3>" { word = "gave" }
"give<V><PART>" { word = "given" }
```

Figure 10: Sample entries of irregular inflections in the English morphologicon

The Italian morphologicon was manually created during the first half of the project but it was a preliminary version, containing only 84 entries and no regular morphology information whatsoever. It has been heavily extended since and now it contains 220 entries and information on the regular inflections of verbs, adjectives, and nouns, as well as information on the concatenation of articles and prepositions and on how articles vary depending on the initial letter of the word they modify. Below there is an example of entries for regular gender and number inflection of adjectives.

```
// Adjectives
// Source: https://www.treccani.it/
enciclopedia/genere-e-numero-degli-
aggettivi_%28La-grammatica-italiana%29/

"e_<JJ><FEM><SG>" { suffix = "e"}
"e_<JJ><MASC><SG>" { suffix = "e"}
"e_<JJ><FEM><PL>" { suffix = "i"}
"e_<JJ><MASC><PL>" { suffix = "i"}

"o_<JJ><FEM><SG>" { suffix = "a"}
"o_<JJ><MASC><SG>" { suffix = "o"}
"o_<JJ><FEM><PL>" { suffix = "e"}
"o_<JJ><MASC><PL>" { suffix = "i"}

"a_<JJ><FEM><SG>" { suffix = "a"}
"a_<JJ><MASC><SG>" { suffix = "a"}
"a_<JJ><FEM><PL>" { suffix = "e"}
"a_<JJ><MASC><PL>" { suffix = "i"}
```

Figure 11: Morpheme entries for regular gender and number adjective inflection in the Italian morphologicon

4.3 Grammar coverage and quality

In addition to the creation of templates for the conceptual structures and the extension of our lexical resources, the other main way of improving our grammar-based generator is to develop new rules or refine pre-existing ones so as to increase the system's coverage and quality.

In this section we report on the improvements to the grammars carried out throughout the project's development, but especially during the second year.

4.3.1 Creation of project-specific rules

In order to provide good coverage for the project's domain, we created some rules specifically designed to fit xR4DRAMA's needs.

First of all, we developed new rules in the first grammar, the one responsible for the mapping between the first layer of representation (conceptual) and the second (semantic). The aim of most of these rules was to transform the concepts used as labels in the project's ontology into the appropriate semantic entities for the specific contexts. We also created a rule to determine when a node corresponds to a named entity.

But the main improvement comes from aggregation rules for sentence packaging. Some rules had already been developed during the first year of the project. For example, at the semantic level, the system looks for structures sharing the same location and merges them together using coordination to create a more complex structure, see Figure 12 below for an example.

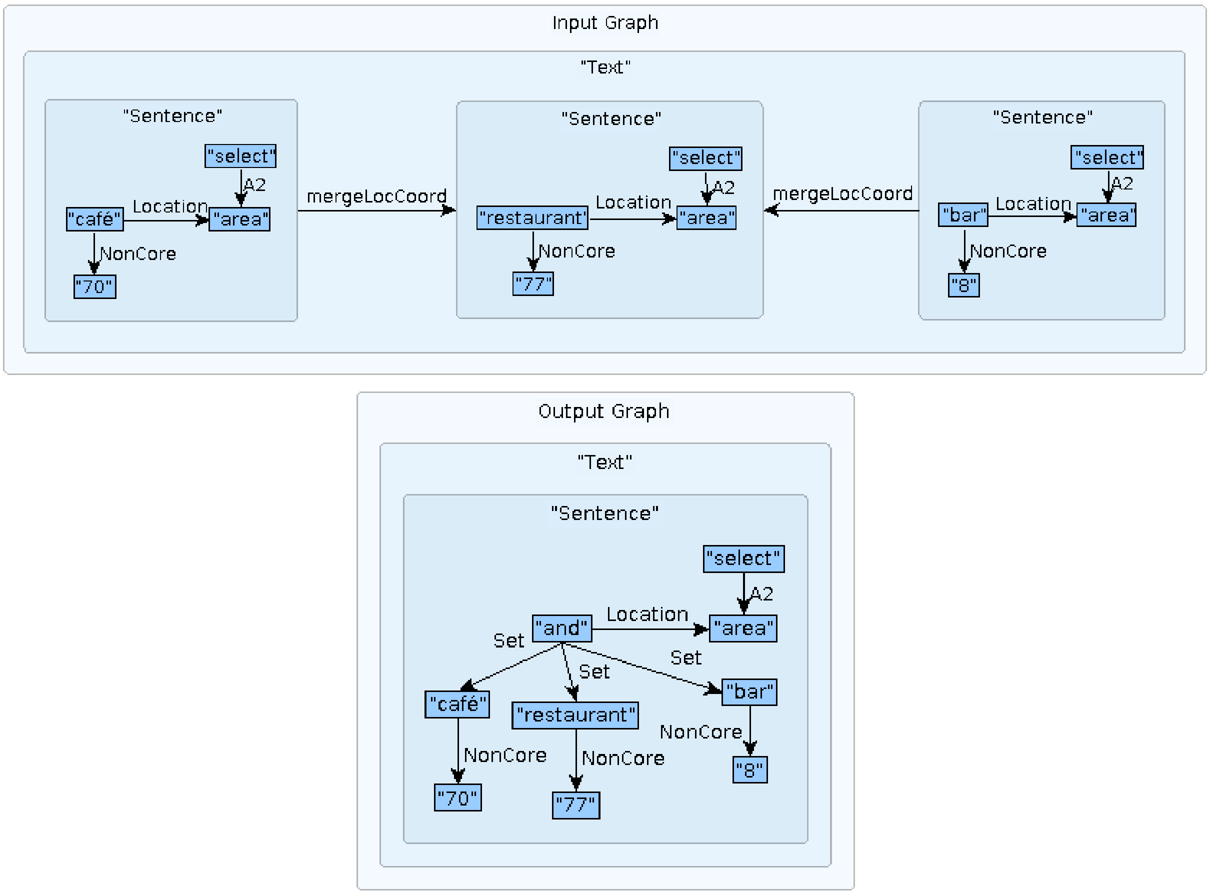


Figure 12: Semantic aggregation rule: three semantic structures are merged into one

In this second half of the project, we developed more rules of this kind. For instance, the one portrayed below (Figure 13), merges together semantic structures that share the same root but have different arguments into a single structure whose root remains the same but has both arguments.

Moreover, we developed rules for contrast aggregation between semantic structures when one of them contained a negation of some kind while the other did not. This was done to improve the coherence of sentences like *“No people, 2 animals and 3 vehicles are in danger.”* which now is realised as *“2 animals and 3 vehicles are in danger, but no people are in danger.”*

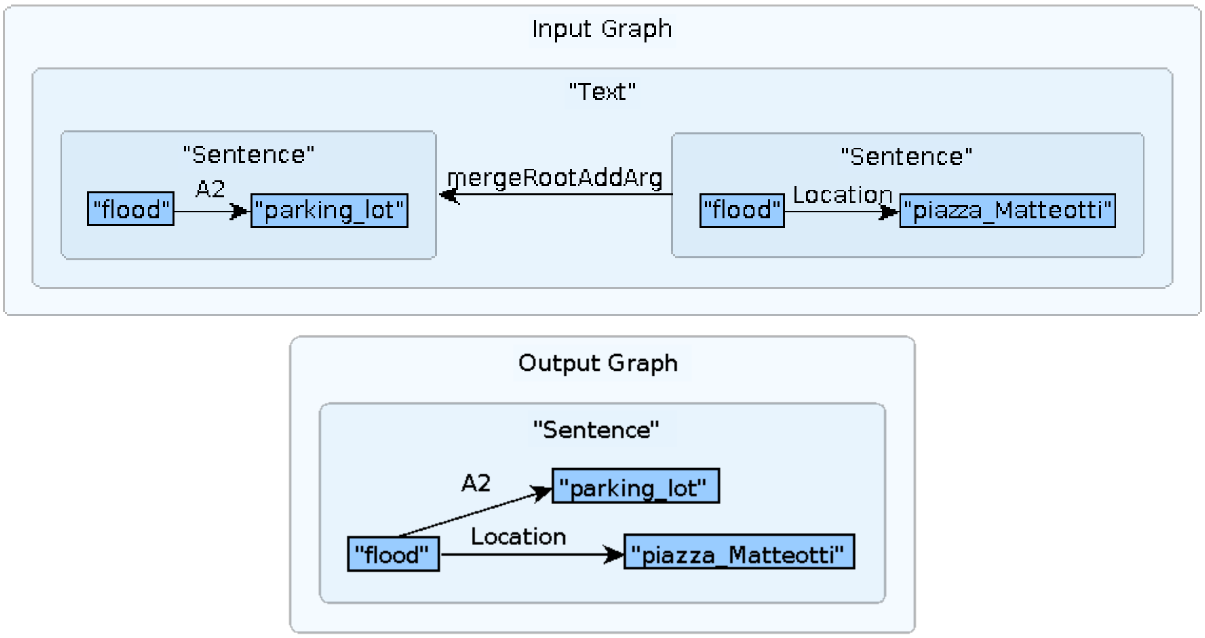


Figure 13: Another aggregation rule: two semantic structures are merged into one

Still at the semantic level, rules for aggregation when exactly the same time information is present in the conceptual structures were created and existing rules were modified to prevent them from applying when time information did not coincide. This mainly affected the timeline reports for PUC1.

At the syntactic level, also new aggregation rules were created especially for this project, although we believe they might be easily extended to other domains. The rules apply to very similar syntactic structures with contrasting semantic content, more specifically, structures whose subjects have opposite polarity. These structures are joined together in a single sentence that connects the two clauses through an adversative coordinating conjunction, as illustrated in the following figure.

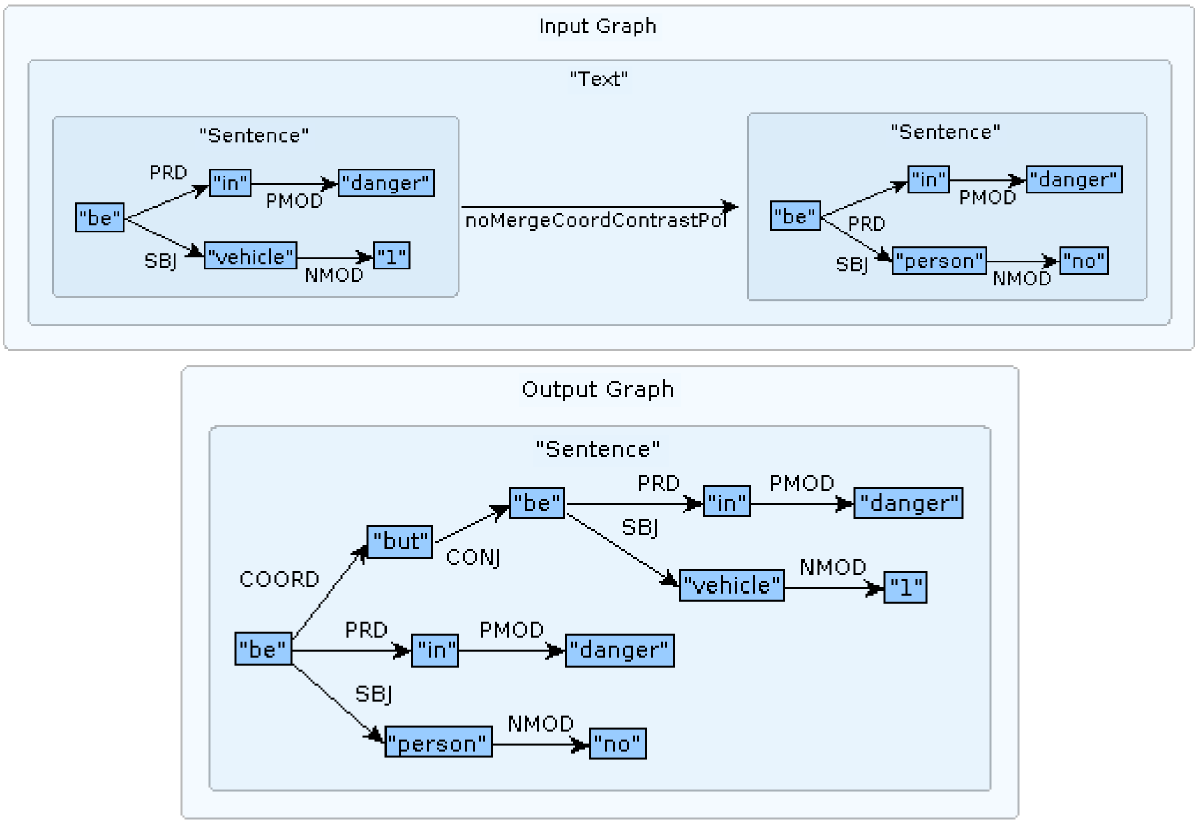


Figure 14: Application of syntactic aggregation rule: two sentences with different polarity subjects are joined via adversative coordination as two clauses in a single sentence

4.3.2 Extensions to language-independent rules

Regarding the improvement of generic rules that apply to all languages, we continued the efforts already started during the first year of development, mainly focusing on word elisions and pronominalisations. These advances increase the fluency and readability of generated texts.

For instance, one of the implemented rules looks for correferent expressions in the subject position, and when it finds two consecutive sentences with correferent subjects, it changes the one in the second sentence for the appropriate pronoun. Figure 15 illustrates this rule, pay special attention to the attributes of the modified node, it maintains the lexical information, but it has been pronominalised. In this case, the second sentence will end up being verbalised as *"It is free"* instead of *"The parking is free"*, given that the subject had already been introduced in the previous sentence *"Valet parking is available in this hotel"*.

Similar rules have been created to cover other kinds of constructions that can also benefit from pronominalisation. See the summary in Section 4.4 to see other examples.

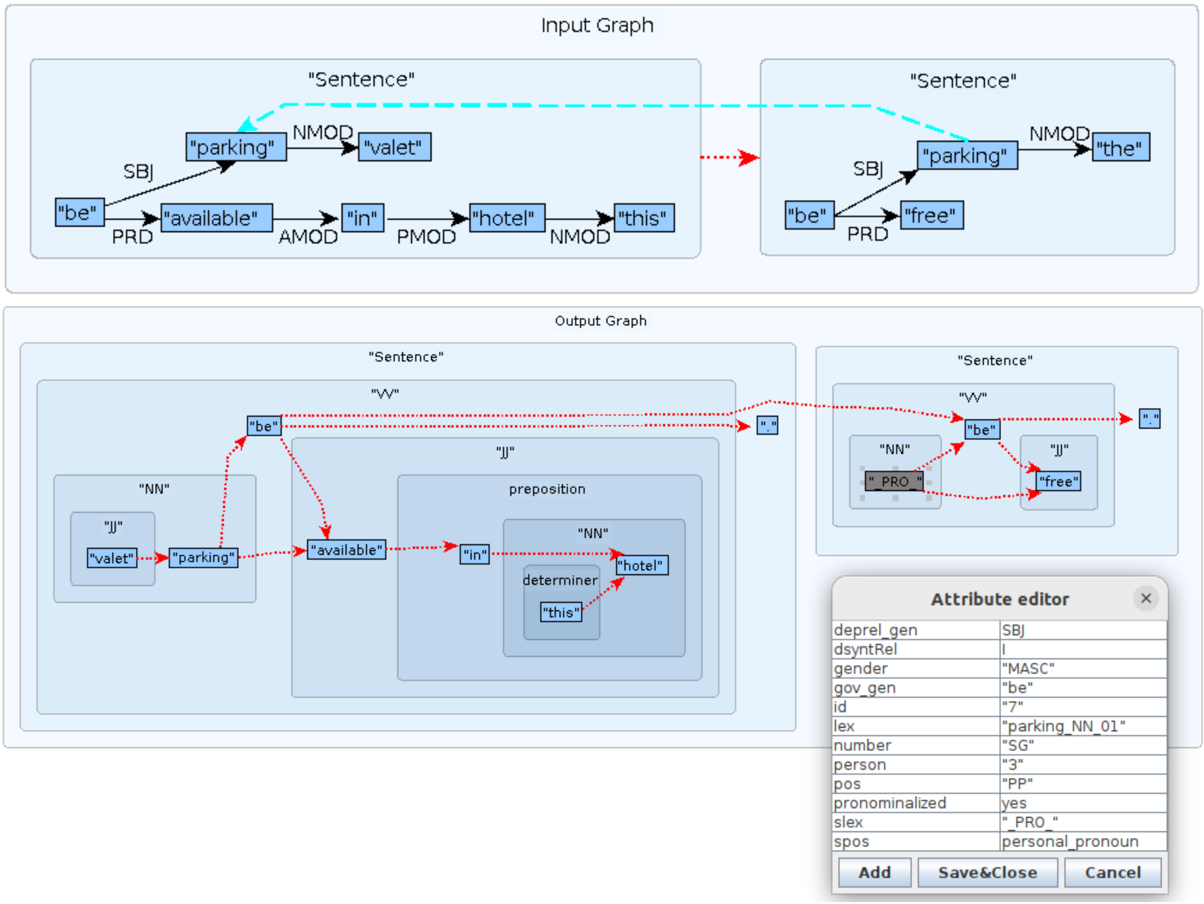


Figure 15: Pronominalisation rule applied to correferent subjects.

In addition to the development of this referring expressions sub-module that takes care of pronominalisations, we have also created rules that introduce elision of unnecessary words, such as repeated subjects and/or verbs in coordinated clauses.

The example below shows the application of both rules for subject and verb elision. We show the result of several steps in the process at once (surface syntactic structure - morphological structure - final sentence) so that the effect of the elision can be seen, since in the step in which it is applied, only an attribute is added to the nodes that will be elided later. Note that the two coordinated clauses in the input graph share the same subject *"a flood"* and the same auxiliary verbs *"have"* + *"be"* + main verb (participle).

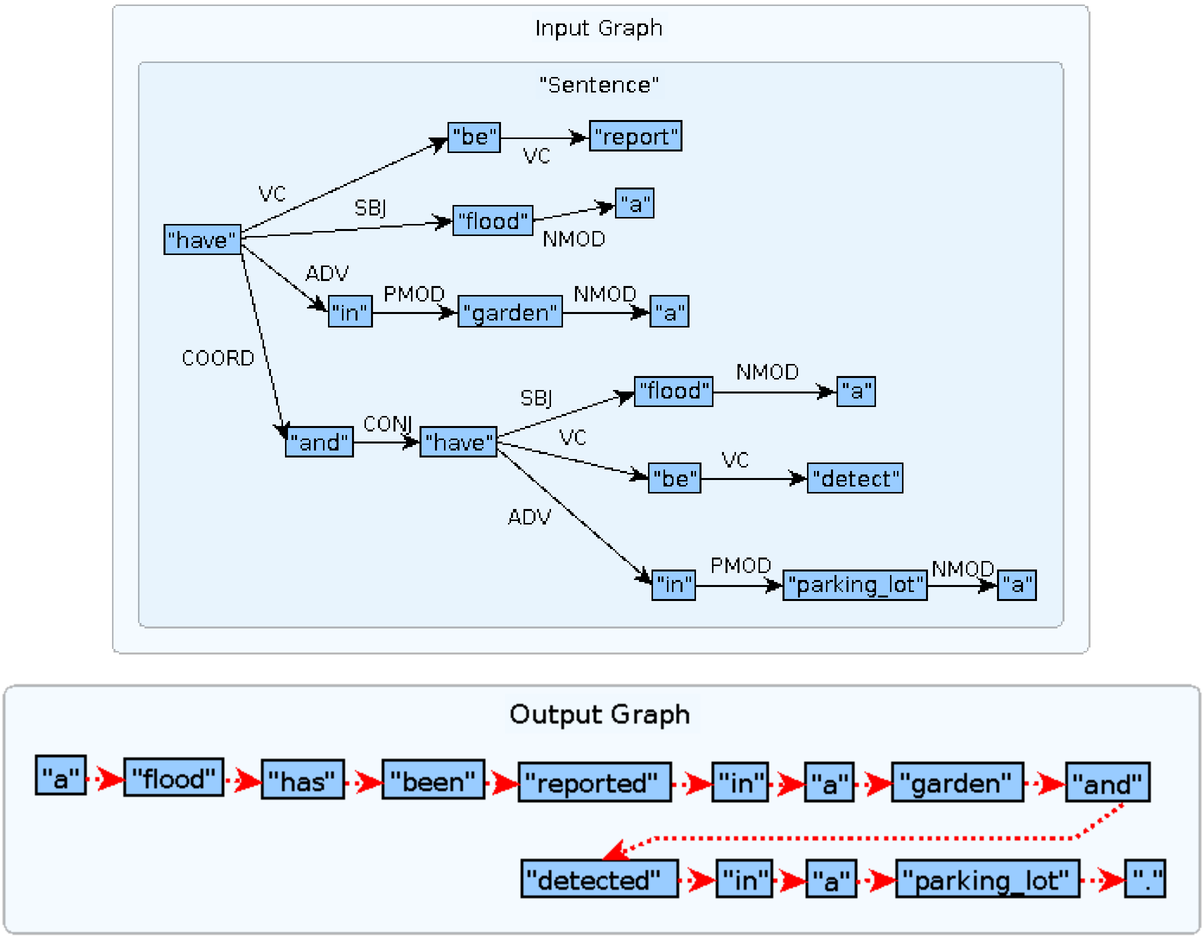


Figure 16: Elision rule applied to repeated subjects and auxiliary verbs in coordinated clauses
"A flood has been reported in a garden and (a flood) (has been) detected in a parking lot"

Rules for the correct linearization of focused adverbial adjuncts were also created. These rules apply, for instance, to specific time information in a timeline report so that instead of having this information at the end of the sentence, it appears at the beginning (see Section 3.1 for an example of such a text).

New rules were introduced to generate sentences for headlines written with a more telegraphic style, like avoiding the introduction of support verbs, for instance. This was useful for this project in the creation of titles for the warnings in PUC1 and for the POIs in PUC2.

A rule was developed to add an adverb expressing the meaning "in addition" to a sentence following another with which it shares the same subject and root at the surface syntactic level. Figure 17 illustrates the application of this rule. It covers sentences that have not been previously aggregated at the semantic level, because the lexical roots of the structures are not the same although their subjects are.

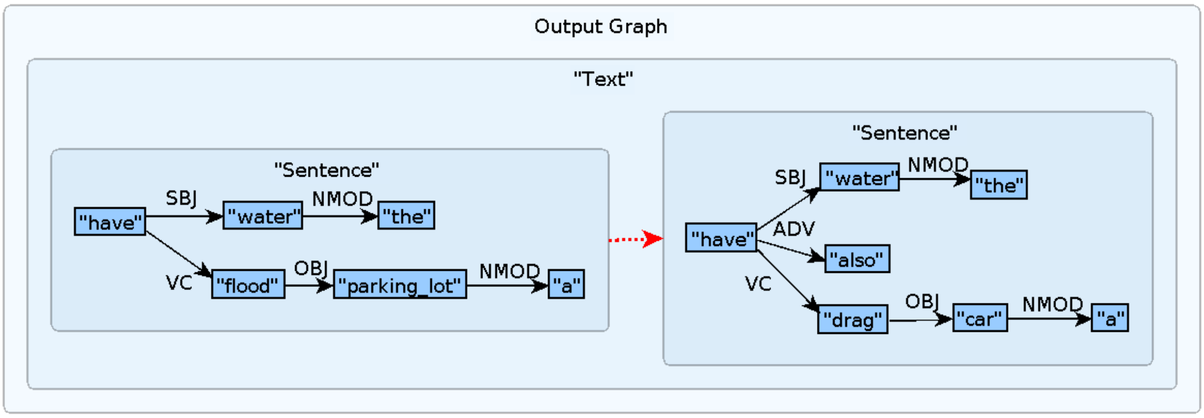


Figure 17: Effect of applying a rule that adds the adverb "*also*" to a sentence following another with the same subject and syntactic root

And finally, as previously mentioned in D3.6, we had implemented a new morphology generation sub-module. We continued this line of work creating the necessary grammar rules and adding changes to the dictionaries so that regular morphological rules are applied whenever possible. New rules mainly covered regular verb morphology, for finite and non-finite verbs. This increases the robustness of the generator to new inputs and reduces the number of entries needed in morphological dictionaries (see Section 4.2 for further details on this).

4.3.3 Extensions to English and Italian rules

The main extension to English specific rules concerned the coverage of interrogative sentences. Although, at this point, it does not seem that the project will be benefiting from these improvements, we thought that the need for communication between the control room and people in the field might arise in the future. Thus, we started working on the implementation of interrogative sentence coverage for English.

Since we had never worked with interrogative sentences before, many new rules were created, mainly for the introduction of "do" support when needed, but also to deal with preposition stranding and word order in these sentences in general, which differs from that of declarative sentences.

In the example below we can see two stages of the generation process for an interrogative sentence. Figure 18 illustrates how the transition from deep syntactic structure to surface syntactic structure is affected when the sentence is interrogative. Notice that "do" support is introduced as the root of the syntactic tree, and also that instead of introducing an indefinite article "a" to the word "danger", the negative polarity item "any" is used, a property that negative and interrogative sentences have in common. Figure 19, on the other hand, shows the result of applying the new linearising rules to the same sentence, so that the word order is the appropriate one. (The final sentence would be *"Do you see any danger for people?"*)

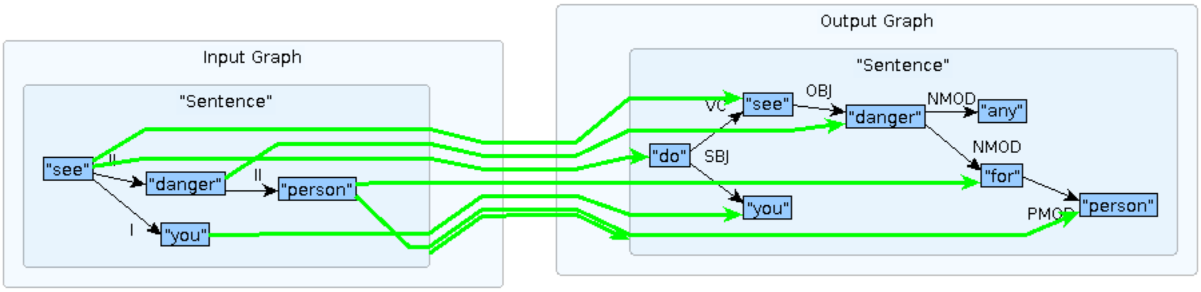


Figure 18: Deep Syntactic to Surface Syntactic structure for an interrogative sentence

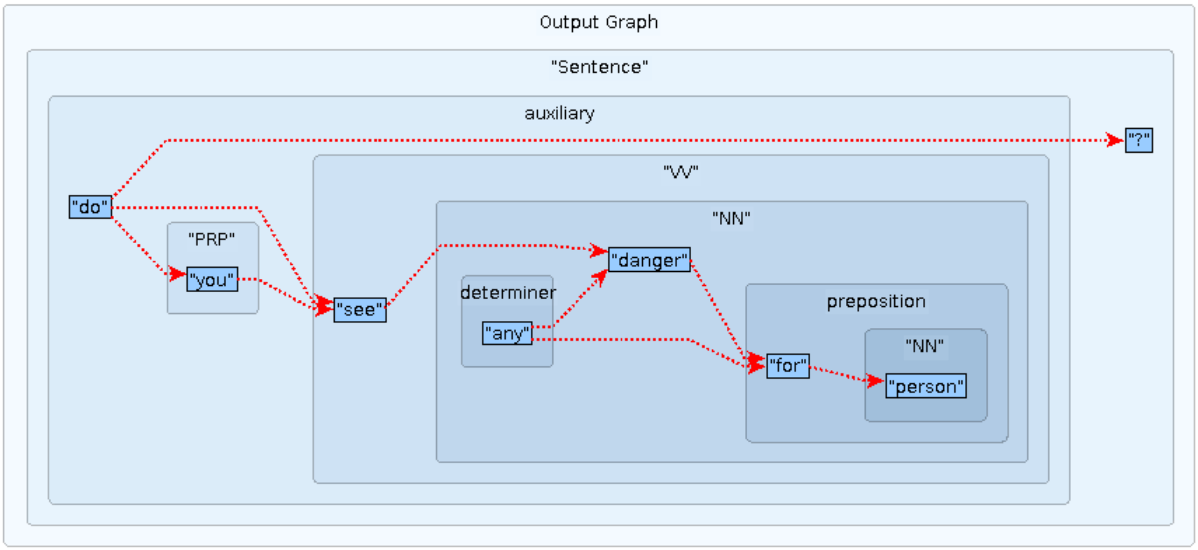


Figure 19: Effect of new word ordering rules for interrogative sentences.

Open questions, such as *"What is the main object which was affected by the flood?"*, are also covered by our text generation module.

As for the extension to the Italian rules, the main improvements concerned existential constructions and perfective constructions both in active and passive voice.

Italian existential constructions require the introduction of a particle, "ci", that functions as an expletive (similar to "there" in English existential constructions). So, new rules were created to insert this particle at the surface syntactic level. We also created and adapted existing rules to deal with the linearisation of such constructions, especially if the existential is negated. The system is now able to generate sentences like *"Ci sono dei ristoranti nell'area selezionata"* or *"Non c'è nessun parcheggio nell'area selezionata"* (There are restaurants in the selected area, there are no parking lots in the selected area).

Perfective constructions in Italian have particular agreement properties: both the participle and the auxiliary "essere" (to be) when the voice is passive need to agree with the subject in number and gender (differently from English but also from other romance languages). We developed rules that insert all the needed auxiliary verbs at the surface syntactic level (see Figure 20) and rules that take care of agreement (with special attention to coordinated subjects) at the transition from surface syntactic to morphologic structures. The following sentences are examples of what is now covered by our generation module in Italian: *"Nessuna emergenza sembra essere stata segnalata"* or *"Un' ostruzione, un'alluvione e un trascinamento sono stati segnalati"* (No emergency seems to have been detected, An obstruction, a flood and debris have been reported).

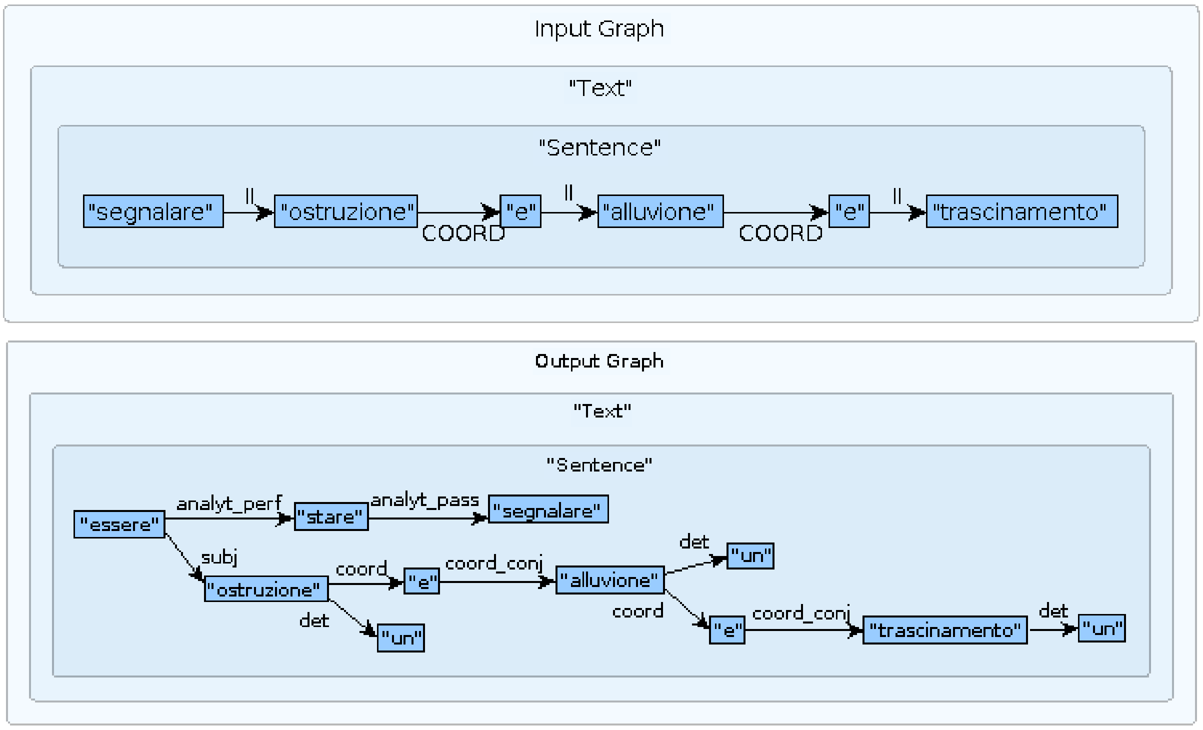


Figure 20: Deep Syntactic to Surface Syntactic structure for a perfective construction in Italian

Finally, we also implemented rules for dealing with other language-specific characteristics such as the change in determiner depending on the word they accompany ("un/uno", "il/lo", for instance), the fact that the presence of a possessive does not preclude the introduction of a determiner (*"la mia macchina"*), or the concatenation that happens between prepositions and determiners (*"nella piazza"*).

4.4 Summary

Table 3 below numerically summarises the development of the grammars during the course of xR4DRAMA's development (M0 - M12 - M24), taking as starting point the generator as reported in November 2020 in the final deliverable of the V4Design project (D5.5, H2020-779962), i.e., just before the start of xR4DRAMA.

	xR4DRAMA M0	xR4DRAMA M12	xR4DRAMA M24
Supported Languages	-	EN	EN, IT
# Rules	1,995	2,147	2,483
% of language - independe nt rules	Con-SMorph (1,995) : 74%	Con-SMorph (2,147) : 75%	Con-SMorph (2,483) : 76%
	- Con-Sem (468) : 97% - Aggregation (309) : 100% - Sem-DSynt (232) : 78% - DSynt-SSynt (645) : 56% - SSynt-DMorph (223) : 53% - DMorph-SMorph (117) : 50%	- Con-Sem (505) : 97% - Aggregation (357) : 100% - Sem-DSynt (243) : 78% - DSynt-SSynt (670) : 56% - SSynt-DMorph (241) : 54% - DMorph-SMorph (131) : 54%	- Con-Sem (571) : 97% - Aggregation (393) : 100% - Sem-DSynt (283) : 81% - DSynt-SSynt (762) : 57% - SSynt-DMorph (327) : 56% - DMorph-SMorph (147) : 56%

Table 3: Overview of the development of the generation grammars

The previously discussed improvements to our generator do not only have an impact on the specific project for which they were developed, but also on one of the most challenging benchmark datasets for structured data-to-text natural language generation. Recall from D3.6 that FORGe was the best system at the WebNLG 2017 task[10] (automatic verbalisation in English of 400 DBpedia properties) according to all human evaluations, and the most portable generator, with the best results for all metrics on unseen data. However, despite achieving very high accuracy and grammaticality, the fluency of the generated texts can be improved, as shown in a more recent large-scale evaluation: WebNLG+2020 (Castro Ferreira et al., 2020).

As a way of visually summarising the advances made during the development of this project, we show how some of the suboptimal outputs from this evaluation have improved thanks to the work carried out these 24 months. The following table illustrates with a couple of examples the impact of our work on this benchmark.

[10] https://webnlg-challenge.loria.fr/challenge_2017/

xR4DRAMA M0	xR4DRAMA M12 (improvements in blue)	xR4DRAMA M24 (further improvements in green)
Atatürk_Monument_(İzmir) , inaugurated on July_27_(1932) , is made of Bronze . It is in Turkey , the leader of which is Ahmet_Davutoğlu . The capital of Turkey is Ankara . The largest city in Turkey is Istanbul . The currency of Turkey is the Turkish_lira .	Atatürk_Monument_(İzmir) , inaugurated on July_27_(1932) , is made of Bronze . It is in Turkey , the leader of which is Ahmet_Davutoğlu . The capital of Turkey is Ankara , the largest city in Turkey Istanbul and the currency of Turkey the Turkish_lira .	Atatürk_Monument_(İzmir) , inaugurated on July_27_(1932) , is made of Bronze . It is in Turkey , whose leader is Ahmet_Davutoğlu , Turkey 's capital is Ankara , the largest city in Turkey is Istanbul and Turkey 's currency is the Turkish_lira .
The Acharya_Institute_of_Technology is in Karnataka . Telangana is located to the northeast of Karnataka . Arabian_Sea is located to the west of Karnataka . The governing body of Tennis is the International_Tennis_Federation .	The Acharya_Institute_of_Technology is in Karnataka . Telangana is located to the northeast of Karnataka and Arabian_Sea is located to the west of Karnataka . The governing body of Tennis is the International_Tennis_Federation .	Acharya_Institute_of_Technology is in Karnataka . Telangana is located to its northeast and Arabian_Sea to its west . Tennis 's governing body is the International_Tennis_Federation .
250_Delaware_Avenue , the architectural style of which is Postmodern_architecture , is in Buffalo_(New_York) . Its construction began in January,_2014 . 250_Delaware_Avenue has a floor area of 30843.8_square_meters . It has 12 floors .	250_Delaware_Avenue , the architectural style of which is Postmodern_architecture , is in Buffalo_(New_York) . Its construction began in January,_2014 . 250_Delaware_Avenue has a floor area of 30843.8_square_meters and 12 floors .	250_Delaware_Avenue , whose architectural style is Postmodern_architecture , is in Buffalo_(New_York) . Its construction began in January,_2014 . 250_Delaware_Avenue has a floor area of 30843.8_square_meters and 12 floors .
William_Anders , whom NASA selected in 1963 , retired on September_01,_1969 . He spent 8820_minutes in space . He was born in British_Hong_Kong on October_17,_1933 . He is a fighter_pilot . He was a crew member of Apollo_8 .	William_Anders , selected by NASA in 1963 , retired on September_01,_1969 . He spent 8820_minutes in space and is a fighter_pilot . He was born in British_Hong_Kong on October_17,_1933 . He was a crew member of Apollo_8 .	William_Anders , selected by NASA in 1963 , retired on September_01,_1969 . He spent 8820_minutes in space , was born in British_Hong_Kong on October_17,_1933 , is a fighter_pilot and was a crew member of Apollo_8 .

Table 4: Improvements on sample raw outputs of the FORGe generator (WebNLG dataset)

5 NEURAL PARAPHRASING EXPERIMENTS

As part of the research on text generation carried out for xR4DRAMA, we conducted some experiments applying a neural text paraphrasing tool to the outputs of FORGe. The idea is to discern whether it is possible to improve the quality of the generated texts by using a hybrid approach: adding more fluency through the neural paraphrasing tool without losing the content accuracy that our rule-based system provides.

5.1 General experiments - WebNLG data

5.1.1 Introduction

The task of paraphrasing consists in transforming an input text so that the output maintains a very similar semantic content but conveyed through different syntactic structures and/or lexical choices. Neural approaches to paraphrasing have successfully been applied to data augmentation in question answering (Dong et al., 2017; Gan and Ng, 2019), machine translation (MT) (Hu et al., 2019; Khayrallah et al., 2020), task-oriented dialog (Niu and Bansal, 2018; Niu and Bansal, 2019), and new MT metrics (Banerjee and Lavie 2005; Zhou et al., 2006; Denkowski and Lavie, 2010; Thompson and Post, 2020a).

5.1.2 Approach

Traditionally, neural paraphrasing approaches make use of Neural Machine Translation (NMT) either directly, by translating the text into another language and then back to the original one, using one or several pivot languages (Mallinson et al., 2017; Aziz and Specia, 2013), or more indirectly, by training on synthetic paraphrase data created by taking bitext and translating one side into the language of the other side (Wieting et al., 2017; Wieting and Gimpel, 2017; Hu et al., 2019). However, the inherent flaw of these techniques is that ambiguities in one language can result in inappropriate paraphrases in the other.

Moreover, paraphrasing text in a way that creates non-trivial differences between input and output is very challenging. Thompson and Post (2020a) show that merely using beam search to generate paraphrases with a multilingual NMT model trained on a large general domain corpus produces trivial copies most of the time. Adding constraints, such as avoiding specific words in the input, helps alleviate this problem.

We adopted the approach introduced by Thompson and Post (2020b), using a general domain multilingual NMT model but discouraging it from generating n-grams present in the input by down-weighting these tokens without completely disallowing them (i.e., adding a softer constraint), which provides outputs lexically biassed away from the inputs, generating non-trivial paraphrases.

More formally, for two sentences *x* and *y*, *M(x)* denotes the meaning of sentence *x*, and *S(x, y)* indicates the lexical and/or syntactic similarity between sentences *x* and *y*. The problem of paraphrasing the sentence *x* is reduced to the task of finding optimal ŷ:

$$\hat{y} = \arg\max_y [p(y|M(x)) - \alpha \cdot S(x; y)]$$

In the equation, α controls the trade-off between the fluency conditioned on semantic similarity and the lexical and/or syntactic diversity.

N-gram overlap algorithm is used in order to calculate *S(x, y)*. It penalises the creation of *n*-grams (up to 4-grams) which are familiar with corresponding ones in the input sequence. At each decoding step, the algorithm checks whether any of the target vocabulary subwords begin the last word of an input n-gram. Penalties are performed to these subwords by subtracting $\alpha \cdot n^{\beta}$ from the output log probabilities before selecting candidates, where α is the aforementioned trade-off, and β is another hyperparameter which is defined as 4 according to experimental results (Thompson and Post, 2020b).

5.1.3 Methodology

The aim of this study was to discover the potential of applying the neural paraphrase model presented above to our grammar-based generation approach. In order to do so, we applied the paraphrasing model to the outcome of UPF's FORGe generator as a post-processing stage.

We applied FORGe to the WebNLG dataset (Castro Ferreira et al., 2020) and randomly selected 120 output texts to be used as input for the paraphrasing model. Given that some of these texts consisted of only one sentence, but others included multiple sentences, we decided to try two different options: (i) applying the model to the text as a whole (text-wise paraphrasing), and (ii) applying it individually to each sentence to then form the text with the paraphrased sentences (sentence-wise paraphrasing). Our insight was that the latter would imply losing some of the context but gaining more focus over individual meanings.

The original texts and its paraphrases were evaluated in terms of fluency, using a 3-point scale (Bad, Average, Good). We further evaluated the quality of texts of different versions of paraphrase models in comparison to the FORGe outcome in terms of erroneously added content, missing content, meaning changed (used another concept or another morphosyntax category such as number or tense), proper noun changed. The results are shown and discussed in the following section.

5.1.4 Results and discussion

Results from the evaluation in terms of fluency can be seen in Table 5. Both text-wise and sentence-wise paraphrasing show a significant quality improvement over the original texts.

Fluency (3-point scale)	FORGe	Text-wise paraphrase	Sentence-wise paraphrase
Good	48.8%	74.38%	87.6%
Average	50.4%	24.79%	10.74%
Bad	0.8%	0.83%	1.65%

Table 5: Results of scoring the fluency of texts

To better illustrate the scoring and the differences between the three kinds of outputs we discuss a couple of examples below:

1. The texts differ in their lexical choice but maintain the same syntactic structure. In this case, both texts were scored as “good”.
   FORGe: i) “AMC Matador is also known as AMC Ambassador.”
   ii) *“(66063) 1998 RO1 was last seen on November 04, 2013.”*
   Paraphrase: i) *“AMC Matador is also referred to as AMC Ambassador."*
   ii) *“(66063) 1998 RO1 was last observed on November 04, 2013.”*
2. The output of FORGe shows an excessive use of the preposition ’of’ which could be avoided by using attributive nouns. In this case, we scored the output of FORGe as “average” and the paraphrase as “good”.
   FORGe: *“The length of the runway of Alpena County Regional Airport is 1,533.”*
   Paraphrase; *“The runway length of Alpena County Regional Airport is 1,533.”*
3. FORGe preserves foreing words that hinder the text’s readability and interpretation.
   FORGe: *“The LCCN number of Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg is 32024459.”*
   Paraphrase: *“The LCCN number of papers from the Mathematical Seminar of the university of Hamburg is 32024459.”*
4. The texts differ in the way the meaning is conveyed by the syntactic structure, being better in one case (scored “average”) than the other (scored “good”).
   FORGe: *“Adonis Georgiadis is in the Deputy Minister for Development, Competitiveness and Shipping.”*
   Paraphrase: *“Adonis Georgiadis serves as the Deputy Minister for Development, Competitiveness and Shipping.”*
5. Texts with multiple sentences are paraphrased differently.
   FORGE: “Trane, which is a subsidiary, was founded in La Crosse, Wisconsin. The total area there is 58.38.”
   Text-wise: “Trane, a subsidiary, opened in La Crosse, Wisconsin, with a total area of 58.38.”
   Sent-wise: *“Trane, which is an affiliate, was established in La Crosse, Wisconsin. The total area is 58.38.”*

FORGE: *"A Loyal Character Dancer is published in the United States. Asian Americans are an ethnic group there."*

Text-wise: *"A Loyal Character Dancer is printed in North America, where Asian Americans form an ethnic group."*

Sent-wise: *"A Loyal Character Dancer is published. Asian Americans are one ethnic group there."*

As mentioned above, we further evaluated the texts' quality in terms of erroneously added content, missing content, meaning changed (used another concept or another morphosyntax category, such as number or tense), proper noun changed. The results are shown in Table 6. Both paraphrasing methods show high percentages of texts with changed meanings and different proper nouns than the original. The text-wise paraphrasing also shows a high percentage of texts with missing content and even some texts with added content.

Type of error	Text-wise paraphrase	Sentence-wise paraphrase
Added content	4.2%	0.8%
Missing content	20.8%	3.3%
Meaning changed	16.7%	19.83%
Proper noun changed	15.8%	13.2%

Table 6: Percentage of difference between the paraphrased text and FORGe's original output

5.1.5 **Conclusions**

Considering the quantitative and qualitative results discussed in the previous section, we came to several observations. We found that text-wise paraphrasing achieves a better fluency by turning the input's short sentences into clauses and combining them to form longer and more complex sentences. However, there are cases when the last sentence gets omitted entirely, resulting in a large percentage of missing content. Sentence-wise paraphrasing diversifies the vocabulary in single sentences and, at the same time, makes them more fluent, however, semantic accuracy over the entire text is somewhat lower.

So, in conclusion, although fluency results show a very promising picture, both paraphrasing methods provide outputs with a high percentage of missing content, changed meanings and different proper nouns used, even including added content in some cases. This means that the use of these hybrid generation methods in favour of the purely rule-based system FORGe should be carefully pondered, considering the specificities and needs of the project, prior to making a decision.

This investigation was part of the master thesis "Exploring Neural Paraphrasing to Improve Fluency of Rule-Based Generation" (Du, 2021).

5.2 Application to xR4DRAMA

As a second step in our research, we applied to xR4DRAMA the hybrid approach explored in the general experiments previously described. Since not much real data was available at this point of the project's development, we carried out a pilot study with a small selection of data created for the prototypes. In this section we discuss the results we obtained, justifying our decision to not include neural paraphrasing into the project's NLG component.

5.2.1 Examples and discussion

Following the considerations suggested by the general experiments' results and considering the project needs, we decided to apply the paraphrasing model to a selection of outputs from FORGe covering both use cases. We tried several combinations of the hyperparameters, α and *β*, and obtained results that ranged from being very similar to the inputs to being unacceptable semantical divergences.

Unfortunately, the results were not very satisfactory. Fluency did not necessarily improve, and we observed instances of added content as well as missing content, several changes in meaning and slight modifications to proper nouns. Let's discuss each of these concerns individually and show some examples.

Fluency and grammaticality

The improvement of text quality in terms of fluency was marginal, at best. As can be seen in the following example, although there are some nice alternative reformulations that maintain a very good level of semantic similarity (in bold), the paraphrased text contains an agreement mistake ("1 vehicles") and a lexical choice that does not fit the register ("persons" is mostly used in juridical texts).

FORGe	*At 10:30 the visual analysis had not detected any emergency but an obstruction had been reported. 2 people and 1 vehicle were in danger. A car and 2 people were blocked at a bridge.*
Paraphrase	*At 10:30 the visual analyses **failed to detect any emergencies** but an obstruction **was** reported. 2 **persons** and **1 vehicles** were **at risk**. 1 car and 2 **persons** were blocked **on** a bridge.*

We found a surprising amount of grammatical and spelling mistakes as well as lexical choices that are synonyms to the original words but are not well fitted to the context. Let's see some more examples:

FORGe	i) *There are 11 toilets but no internet access points in the selected area.* ii) *The visual analysis has detected a flood in a bar and an urban canal.* iii) *An obstruction and a flood have been reported.* iv) *No people are in danger but 6 vehicles are in danger.*
Paraphrase	i) ***Have** 11 toilets but no **web** access points in the **targeted** area.*

ii) ***Vizual*** *analyses have detected* ***inundation in bar and urban channel.***
iii) *An obstruction and a flood* ***has*** *been reported.*
iv) *No-one is* ***endangered*** *but* ***6-vehicle is endangered****.*

Added or missing content

We found a couple of instances of added content as well as missing content, even though the dataset used was pretty small. These kinds of mistakes are worrisome for a project like xR4DRAMA, in which the veracity of the information provided is of crucial importance.

Examples of additional content:

FORGe i) *At 10:30 the visual analysis had not detected any emergency but an obstruction had been reported.*
ii) *The visual analysis has detected a flood at an urban canal. The river has overflowed.*

Paraphrase i) *By 10:30 visual analyses had detected no emergencies but* ***they*** *had reported obstruction.*
ii) *Visual analyses have detected inundation in* ***one of the city's*** *canals. The* ***city's*** *river is overflowing.*

In the first example, the paraphrased text implies that the obstruction has been reported by visual analysis, which is not true, since reports come from messages sent by citizens. In the second example it is implied that the canal is in a city, which is not necessarily true.

Examples of missing content:

FORGe *Between 1 and 2 people and between 1 and 3 vehicles are in danger but no animals are in danger. The visual analysis has detected a flood at an urban canal. The river has overflowed. An obstruction and a flood have been reported.* ***Some people have been blocked at a house and a tree and a branch have blocked 2 people. Via santo domenico has been flooded. The water has flooded a car and a bike.***

Paraphrase *Between one and two people and between one and three vehicles are endangered but none of the animals are endangered. Visual analyses have detected flooding in an urban channel. The river is overflowing. Obstruction and flooding have been reported.*

In this example important information has been omitted.

Semantic divergences

We have already seen, in some of the previous examples, instances of slight changes of meaning (“in danger” vs. “endangered” or “canal” vs. “channel”), but let’s show other examples:

FORGe i) *A schoolbus has been blocked. A parking lot has been flooded.*
ii) *The water has flooded a car and a bike.*
iii) *Wi-Fi and few plugs are available in this café. The charging switches are located beneath the table.*

Paraphrase i) *One school bus **hasn't been checked**. One parking garage **hasn't been submerged**.*
ii) *The **sea** has inundated one car & one bike.*
iii) *Wi-Fi, and a fewplugs are made available at this **pub**. The charging **switches** are located beneath the table.*

It is important to underline that the changes of meaning may result in untrue statements. Notice that in some of the examples the paraphrased text has a very different meaning, or even the opposite meaning.

Proper noun modifications

There were only a couple of proper nouns in the selected examples, but one of them was slightly modified by the paraphrasing models for all the tested values of the parameter.

FORGe *Via santo domenico has been flooded.*

Paraphrase ***Via "santo dominico"** has been flooded.*

The modification may not seem very important, and in this case, it is easily interpretable. However, we cannot ensure that this will always be the case and changing the name of a street may create unnecessary uncertainties for people in the control room or in the field.

5.2.2 Final decision

In the end, the best overall results, i.e., those with less grammatical errors and better accuracy with respect to the original content of the texts were those obtained when the trade-off parameter favoured semantic similarity over lexical diversity. However, these texts did not contribute any real benefit to the final output (merely changing some words that in most cases fitted the context poorly - such as "endangered" instead of "in danger").

So, considering the results of these experiments and the fact that truthfulness is crucial in a project whose aim is to improve situation awareness for disaster management and media production planning, we decided against applying neural paraphrasing methods as part of xR4DRAMA's text generation module.

6 EVALUATION

Evaluation is a crucial part of the development of text generation systems. However, it is a highly complex task: deciding whether a text actually contains all the information that it is supposed to contain, assessing if it is well written and/or easy to read is by no means a trivial problem. The main strategy in the field has been to create human references for each input, and then compare word-to-word the text generated automatically with these references. However, there are many possible ways to express the same thing, a generated text that doesn't match the reference at all can still be a very good candidate. Synonymy and embeddings have more recently been used to relax the matches between reference and predicted texts, but this is still not satisfactory. In the end, the safest and best way to evaluate text is to resort to human evaluators, even though this solution is costly in terms of setup and time.

Unfortunately, we encountered a serious issue for the evaluation of xR4DRAMA's generation module. The lack of large enough amounts of real data prevented us from running a comprehensive evaluation. However, we performed a preliminary human evaluation of a representative selection of generated texts (from manually crafted data) to demonstrate how the generator performs in the context of this project and get some qualitative feedback.

We selected 10 English texts, 5 for each use case: 4 warning reports and 1 timeline report for PUC1 and 4 short POI's titles and descriptions and a preliminary report for PUC2. We set up a Google form to collect grammaticality, readability and usefulness ratings on a 5 point Likert scale from human volunteer evaluators. The first page of the form is shown below. It contained detailed instructions for the evaluation and an introduction about the project and its use cases.

> **General introduction: please read carefully!**
>
> This task's purpose is to evaluate the 10 provided texts according to their quality in terms of the criteria given in the statements below.
>
> The texts are generated automatically in the context of xR4DRAMA, a project whose objective is to create and enhance situation awareness for those who are, remotely as well as directly, involved in the planning of and the dealing with events and incidents in a specific location. The project has been applied to two different use cases: disaster management (PUC1) - to help the work both of managers in the control room and first responders in the field (at the city of Vicenza, Italy), and media production planning (PUC2) - to support and facilitate the planning activities that need to be carried out by the production management team to create a production plan for a shoot at a specific location (Corfu Island, Greece).
>
> For the disaster management use case, there are two kinds of texts: (i) brief warnings generated when an emergency is detected by visual analysis of images captured through the platform and/or by text analysis of messages sent by citizens, and (ii) timeline reports with a summary of what has happened during a specified period of time.
>
> For the media production use case, there are also two kinds of texts: (i) a preliminary report about the number of facilities available at the region of interest, and (ii) short descriptions of specific facilities available at a determined geo-located point of interest (POI) within the region.
>
> Each text needs to be assessed with respect to the three following criteria (the criteria will be repeated at the top of each text):

- Grammaticality: The text is free of grammatical and spelling errors.
- Readability: The text is easy to read and understand, and it flows well.
- Usefulness: I find the information provided in the text useful for a potential user of this platform.

Please score each text by stating how strongly you agree or disagree with each statement: 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree

You can also leave open feedback for each report, to tell us more about what you liked and didn't like.

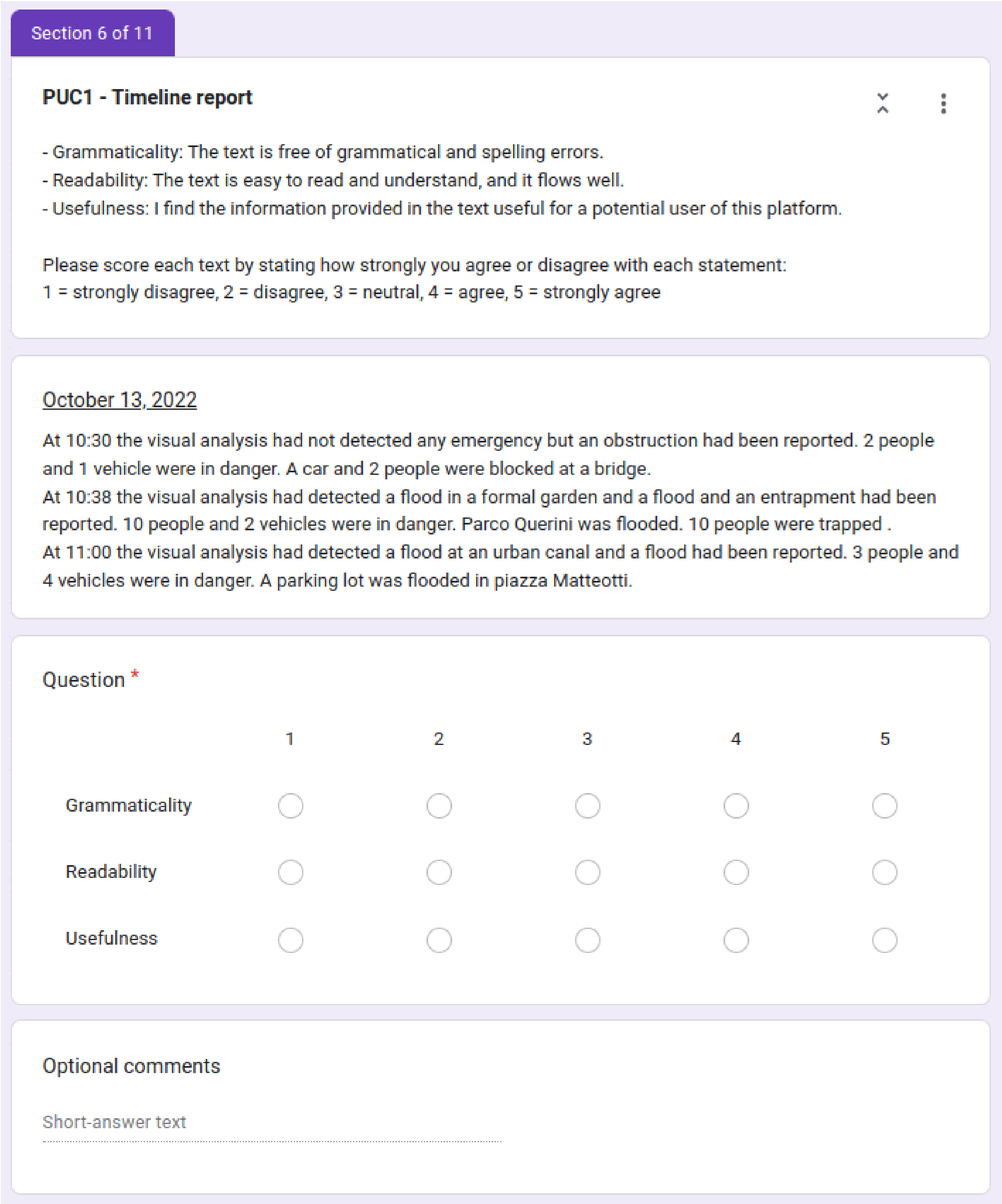
The following 10 pages contained a text each. A screenshot of the evaluation interface for the timeline report for PUC1 is provided in Figure 21.

So far, 5 people have performed the evaluation of the 10 texts according to the three criteria (150 individual ratings in total). Table 7 shows the mean results, which are quite good in general, all over 4 out of 5, grammaticality being the highest.

Grammaticality	Readability	Usefulness
4.74	4.34	4.54

Table 7: Evaluation results (maximum rating = 5)

Considering the qualitative comments left by the evaluators, we believe that the reason for readability having the lowest scores is related to how sentence packaging is performed for texts in PUC1, since it can result in syntactically complex sentences. In fact, the qualitative feedback was extremely useful to understand the reasons behind quantitative evaluation and provided constructive criticism that will undoubtedly contribute to the improvement of the generation module.



Section 6 of 11

PUC1 - Timeline report

- Grammaticality: The text is free of grammatical and spelling errors.
- Readability: The text is easy to read and understand, and it flows well.
- Usefulness: I find the information provided in the text useful for a potential user of this platform.

Please score each text by stating how strongly you agree or disagree with each statement:
1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree

October 13, 2022

At 10:30 the visual analysis had not detected any emergency but an obstruction had been reported. 2 people and 1 vehicle were in danger. A car and 2 people were blocked at a bridge.
At 10:38 the visual analysis had detected a flood in a formal garden and a flood and an entrapment had been reported. 10 people and 2 vehicles were in danger. Parco Querini was flooded. 10 people were trapped .
At 11:00 the visual analysis had detected a flood at an urban canal and a flood had been reported. 3 people and 4 vehicles were in danger. A parking lot was flooded in piazza Matteotti.

Question *

	1	2	3	4	5
Grammaticality					
Readability					
Usefulness					

Optional comments

Short-answer text

Figure 21: Screenshot of the evaluation form

7 CONCLUSIONS

In this report we have given a detailed account of the progress attained throughout the 24 months of the project's development regarding the task of multilingual text generation in WP3, paying particular attention to the work carried out over the last 12 months.

The text generation module has been developed as expected. Two different kinds of target texts for each pilot use case have been determined to address users' needs, the final interface with the KB has been defined, lexical resources have been extended and the module's coverage has been significantly enhanced. The main improvements made to UPF's multilingual discourse generator are the following:

- Final predicate-argument templates for conceptual structures and the necessary lexical resources were manually crafted to cover xR4DRAMA's domain.
- Grammars were augmented to improve the system's general coverage, especially for English and Italian, as well as to cover project-specific needs. The total number of rules has increased from 1,995 at the beginning of the project to 2,483 now. As a consequence of all this, generated texts are now more fluent.
- The newly implemented morphology sub-module has been refined and simplified so that morphological regularities are used to the system's advantage.
- The grammar's updates have continued to result in quality improvements of the English generator on WebNLG, one of the most challenging benchmarks in NLG.
- The rule generalisation continued to be improved, grammars were made more language independent: now, 76% of the rules are language-independent, as opposed to 74% at the beginning of the project.

The provided functionalities are in line with the project's timeline. The NLG component has been integrated into the project's general system and minor changes are planned until the end of the project. A preliminary human evaluation has been performed (both in quantitative and qualitative forms) and the suggested improvements will be addressed toward the final version of the system.

In addition, experiments on the combination of our grammar-based method with neural paraphrasing models have been conducted. Unfortunately, the results have not been satisfactory enough to implement the paraphrasing sub-module as part of the text generation component.

Four publications in the context of xR4DRAMA were produced:

Kasner, Z., S. Mille and O. Dušek (2021). Text-in-Context: Token-Level Error Detection for Table-to-TextGeneration. In *Proceedings of the 14th International Conference on Natural Language Generation*, pp. 259-265, Aberdeen, UK (Online). **pdf**

Mille, S., T. Castro-Ferreira, A. Belz and B. Davis (2021). Another PASS: A Reproduction Study of the Human Evaluation of a Football Report Generation System. In *Proceedings of the 14th International Conference on Natural Language Generation*, pp. 286-292, Aberdeen, UK (Online). **pdf**

Mille, S., Dhole, K.D., Mahamood, S., Perez-Beltrachini, L., Gangal, V., Kale, M., van Miltenburg, E. and Gehrmann, S. (2021). Automatic Construction of Evaluation Suites for Natural Language Generation Datasets. In *Proceedings of the Thirty-Fifth Annual Conference on Neural Information Processing Systems, Datasets and Benchmarks Track (Round 1)*. In Press.

Pérez-Mayos, L., A. Táboas García, S. Mille and L. Wanner (2021). Assessing the Syntactic Capabilities of Transformer-based Multilingual Language Models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3799-3812. Association for Computational Linguistics. **pdf**

8 REFERENCES

Aziz, W. and Specia, L. 2013. *“Multilingual WSD-like constraints for paraphrase extraction”*, In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, p. 202–211.

Banerjee, S. and Lavie, A. 2005. *“METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”*, In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, p. 65–72.

Bohnet, B. and Wanner, L. 2010. *"Open source graph transducer interpreter and grammar development environment"*, In Proceedings of the Seventh International Conference on Language Resources and Evaluation, p.211-218.

Denkowski, M. and Lavie, A. 2010. *“Extending the METEOR machine translation evaluation metric to the phrase level”*, In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, p. 250–253.

Dong, L., Mallinson, J., Reddy, S. and Lapata, M. (2017). *“Learning to paraphrase for question answering”*, arXiv preprint arXiv:1708.06022.

Du, S. (2021). *“Exploring Neural Paraphrasing to Improve Fluency of Rule-Based Generation”*, Master thesis on Intelligent Interactive Systems. Universitat Pompeu Fabra.

Ferreira, T., Gardent, C., Ilinykh, N., Van der Lee, C., Mille, S., Moussallem, D. and Shimorina, A. 2020. *"The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (WebNLG + 2020)"*, In Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+).

Gan, W. C. and Ng, H. T. (2019). *“Improving the robustness of question answering systems to question paraphrasing”*, In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, p. 6065–6075.

Hu, J. E., Singh, A., Holzenberger, N., Post, M. and Van Durme, B. 2019. *“Large-scale, diverse, paraphrastic bitexts via sampling and clustering”*, In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), p. 44–54.

Khayrallah, H., Thompson, B., Post, M. and Koehn, P., 2020. *“Simulated multiple reference training improves low-resource machine translation”*, In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 82-89.

Mel’čuk, I.A., 1988. *“Dependency syntax: theory and practice”*, SUNY press.

Mille, S., Dasiopoulou, S. and Wanner, L., 2019. *"A portable grammar-based NLG system for verbalization of structured data"*, In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 1054-1056.

Niu, T. and Bansal, M., 2019. *"Automatically Learning Data Augmentation Policies for Dialogue Tasks"*, In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), p. 1317-1323.

Rambow, O. and Korelsky, T. 1992. *"Applied text generation"*, In Proceedings of ANLP, ACL, pp.40-47.

Thompson, B. and Post, M. 2020a. *"Automatic machine translation evaluation in many languages via zero-shot paraphrasing"*, In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 90-121.

Thompson, B. and Post, M. 2020b. *"Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity"*, In Proceedings of the Fifth Conference on Machine Translation, p. 561-570.

Wieting, J. and Gimpel, K. 2017. *"Pushing the limits of paraphrastic sentence embeddings with millions of machine translations"*, arXiv preprint arXiv:1711.05732.

Wieting, J., Mallinson, J. and Gimpel, K., 2017. *"Learning paraphrastic sentence embeddings from back-translated bitext"*, In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 274-285.

Zhou, L., Lin, C.-Y., Munteanu, D. S. and Hovy, E. 2006. *"Paraeval: Using paraphrases to evaluate summaries automatically"*, In Proceedings of the human language technology conference of the NAACL, main conference, p. 447–454.

A Appendix

A.1. Input JSON file for PUC1


```
{
 "header": {
   "first_timestamp": "2022-03-10T09:58:15.867Z",
   "last_timestamp": "2022-03-23T19:58:15.867Z",
   "created_at": "2022-03-24T13:59:49.063Z",
   "id": "1259759244XPDIXHErcHNAwyQyOxbM",
   "project": [
     {
       "project_id": "citizen",
       "text": [
         {
           "data": {
             "timeReference": null,
             "tags": [
               "Humans in Danger",
               "Building in Danger"
             ],
             "type": "incident",
             "situations": [
               {
                 "label": "obstruction",
                 "agents": [],
                 "affected_objects": [
                   {
                     "name": "person",
                     "quantity": "some"
                   }
                 ],
                 "location": "house"
               },
               {
                 "label": "flood",
                 "agents": [],
                 "affected_objects": [
                   {
                     "name": "via santo domenico",
                     "quantity": "1"
                   }
                 ]
               }
             ],
             "category": "Disaster Management",
             "subcategory": "Humans in Danger"
           },
           "meta": {
             "type": null,
             "date": null,
             "times": null,
             "id": "123e4567-e89b-42d3-a456-556642440000",
             "entity": null,
             "location": null,
             "sourceText": null,
             "sourceType": null,
```

```
        "timestamp": "1666781637",
        "project_id": "MyProject"
      }
    },
    {
      "data": {
        "timeReference": null,
        "tags": [
          "Humans in Danger",
          "Building in Danger"
        ],
        "type": "incident",
        "situations": [
          {
            "label": "obstruction",
            "agents": [
              "tree",
              "branch"
            ],
            "affected_objects": [
              {
                "name": "person",
                "quantity": "2"
              }
            ]
          },
          {
            "label": "flood",
            "agents": [
              "water",
              "mud"
            ],
            "affected_objects": [
              {
                "name": "car",
                "quantity": "1"
              },
              {
                "name": "bike",
                "quantity": "1"
              }
            ]
          }
        ],
        "category": "Disaster Management",
        "subcategory": "Humans in Danger"
      },
      "meta": {
        "type": null,
        "date": null,
        "times": null,
        "id": "123e4567-e89b-42d3-a456-556642440000",
        "entity": null,
        "location": null,
        "sourceText": null,
        "sourceType": null,
        "timestamp": "1666706575",
        "project_id": "MyProject"
      }
```

```
      }
    ],
    "visuals": {
      "header": {
        "timestamp": "2022-10-13 12:25:24.174139",
        "sender": "Visual Analysis",
        "entity": "citizen_report_video",
        "simmoid": "d2468739-a34c-4e2b-a791-25489bdffd23",
        "project_id": "citizen"
      },
      "shotInfo": [
        {
          "shotIdx": 0,
          "startFrame": 0,
          "endFrame": 0,
          "objectsFound": [
            {
              "type": "earth",
              "probability": 0.40424255235004125
            },
            {
              "type": "building",
              "probability": 0.33559939177809445
            },
            {
              "type": "water",
              "probability": 0.03850481054433133
            },
            {
              "type": "sky",
              "probability": 0.10045207525041423
            }
          ],
          "peopleInDanger": 1,
          "vehiclesInDanger": 3,
          "animalsInDanger": 0,
          "riverOvertop": false,
          "infraInDanger": [],
          "objectsInDanger": [],
          "category": "Disaster Management",
          "subcategory": "Flooded Reports",
          "coordinate": [
            40.60165610798415,
            22.79562953003058
          ],
          "area": "canal_urban",
          "areaProb": 0.2858,
          "outdoor": true,
          "emergencyType": "flood",
          "emergencyProb": 0.98
        },
        {
          "shotIdx": 0,
          "startFrame": 0,
          "endFrame": 0,
          "objectsFound": [
            {
              "type": "earth",
              "probability": 0.40424255235004125
```

```
                    },
                    {
                      "type": "building",
                      "probability": 0.335599391778094445
                    },
                    {
                      "type": "water",
                      "probability": 0.03850481054433133
                    },
                    {
                      "type": "sky",
                      "probability": 0.10045207525041423
                    }
                  ],
                  "peopleInDanger": 2,
                  "vehiclesInDanger": 1,
                  "animalsInDanger": 0,
                  "riverOvertop": true,
                  "infraInDanger": [],
                  "objectsInDanger": [],
                  "category": "Disaster Management",
                  "subcategory": "Flooded Reports",
                  "coordinate": [
                    40.60165610798415,
                    22.79562953003058
                  ],
                  "area": "canal_urban",
                  "areaProb": 0.2858,
                  "outdoor": true,
                  "emergencyType": "flood",
                  "emergencyProb": 0.6
                }
              ]
            }
          }
        ]
      }
    }
```


A.2. Input JSON File PUC2


```
{
 "data" : {
   "utilities" : [ {
     "type" : "Parking",
     "location" : null,
     "quantity" : null,
     "qualities" : [ ],
     "relative_position" : null
   }, {
     "type" : "Wifi",
     "location" : null,
     "quantity" : null,
     "qualities" : [ ],
     "relative_position" : null
   } ],
   "coordinates" : [ 39.394867, 20.021863 ],
```

```
    "name" : "Meltemi Beach Bar",
    "description" : "All day beach bar",
    "type" : "logistics",
    "category_name" : "Beer Bar"
  },
  "meta" : {
    "type" : null,
    "date" : null,
    "times" : null,
    "id" : null,
    "entity" : null,
    "location" : null,
    "sourceText" : null,
    "sourceType" : null,
    "timestamp" : null,
    "project_id" : null
  }
}
```