EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 36 months

D5.4: Final technology for multilingual and self-explainable news generation (T5.1)

Executive summary
This deliverable describes the final parts of work conducted within Task T5.1 of Work Package WP5. In the previous Deliverable D5.2 from the same task we described the multilingual language generation technology and demonstrated its flexibility with respect to datasets and domains. In this deliverable, we increase the flexibility with respect to languages. We describe a method to translate a significant proportion of the system itself — rather than the system's *output* — by taking advantage of neural translation models. The technique described in this deliverable simplifies the process of extending the language support of the system, either to completely new languages, or to new datasets in an existing language. We have widened language support in our Eurostat case study system significantly, and the system currently produces text in English, Finnish, Russian, Estonian, Croatian and Slovene for at least one dataset each.

Partner in charge: UH

Project co-funded by the European Commission within Horizon 2020 Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D5.4**
Deliverable full title:	**Final technology for multilingual and self-explainable news generation**
Deliverable short title:	**Final technology for news generation**
Document identifier:	**EMBEDDIA-D54-FinalTechnologyForNewsGeneration-T51-submitted**
Lead partner short name:	**UH**
Report version:	**submitted**
Report submission date:	**31/10/2021**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Leo Leppänen (UH)**
Co-author(s):	**Hannu Toivonen (UH), Michele Boggia (UH)**
Status:	**__ draft, __ final, _x_ submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
01/09/2021	v0.1	Leo Leppänen (UH)	Document initialized.
28/09/2021	v0.2	Michele Boggia (UH)	Section on neural translation.
03/10/2021	v0.3	Leo Leppänen (UH)	Details on translation tool.
04/10/2021	v0.4	Leo Leppänen (UH)	Minor modifications.
08/10/2021	v0.5	Hannu Toivonen (UH)	Removal of old texts, editing.
11/10/2021	v0.6	Leo Leppänen (UH)	Details regarding improved language support.
11/10/2021	v1.0	Leo Leppänen (UH)	Submitted to internal review.
14/10/2021	v1.1	Antoine Doucet (ULR); Matthew Purwer (QMUL)	Internal review.
19/10/2021	v1.2	Leo Leppänen (UH)	Addressed review comments.
19/10/2021	v1.3	Leo Leppänen (UH)	Added recap of D5.2 per review.
21/10/2021	v1.4	Michele Boggia (UH)	Addressed review comments.
21/10/2021	v2.0	Leo Leppänen (UH)	Ready for quality management.
25/10/2021	v2.1	Nada Lavrač (JSI)	Quality control.
26/10/2021	v2.2	Leo Leppänen (UH)	Final modifications.
28/10/2021	final	Leo Leppänen (UH)	Ready for submission.
29/10/2021	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

1 Introduction

The main objective of the EMBEDDIA project is to develop methods and tools for effective exploration, generation and exploitation of online content across languages thereby building the foundations for the multilingual next generation internet, for the benefit of European citizens and industry using less-represented European languages. One facet of this effort is EMBEDDIA work package 5 (WP5), which is concerned with Natural Language Generation (NLG). In order to support journalists and media companies in efficiently reaching as many demographics as possible, the objective of WP5 is to design and develop news automation systems that are transferable across languages, transferable across domains, and transparent in the NLG process; in particular, with the output of NLG that is dynamic, has narrative structures, and uses figurative and colourful language.

More specifically, WP5 aims to develop a flexible, accurate, and transparent NLG system architecture that can be transferred to new domains and languages with minimal human effort; develop tools for creation of dynamically evolving content, incorporating narrative structure and user knowledge; and develop tools for creation of figurative language and headlines. The work package consists of three primary tasks:

- Task T5.1, Multilingual text generation from structured data. The objective of T5.1 is to adapt NLG technology for the requirements of news generation. The task will develop mechanisms for (i) determining what is interesting or important in the given data and deciding what to report, and for (ii) rendering that information in an accurate manner (iii) in multiple languages.
- Task T5.2, Multilingual storytelling and dynamic content generation. The objective of T5.2 is to develop a novel method for automatically organising news articles based on the domain of the article.
- Task T5.3, Creative language use for multilingual news and headline generation. The objective of T5.3 is to make the generated texts more varied and colourful by generating creative expressions, especially in headlines by finding similar terms and metaphors by finding analogous terms in different contexts using context-dependent embeddings. A special focus is on cross-cultural metaphors.

In this deliverable, we report on the final developments in task T5.1. In the previous deliverable D5.2 we defined and introduced automated journalism, the general process of automatically generating news text; we briefly described the state of the art in natural language generation, the technical process by which said news texts are generated; we conducted an analysis of how various approaches to NLG relate to the requirements imposed by the news domain; we described briefly the general natural language generation architecture employed within the EMBEDDIA project; and we described two instantiations of said architecture in the form of two case studies, highlighting how the architecture can be applied to different types of news generation contexts. For more details, please refer to Deliverable D5.2.

Section 2 of this deliverable provides a very brief summary of the parts of Deliverable D5.2 relevant for understanding the work described herein. In this deliverable, we complement the previous work by extending it to support more languages (Section 3), and with a novel approach that uses neural models, more specifically contextual word embeddings as developed in WP1, to help add new languages to the architectures (Section 4). We briefly touch on forthcoming evaluation of the work (Section 5), and then provide our concluding thoughts about this work (Section 6). Finally, we list the associated outputs (Section 7).

2 The EMBEDDIA news generation technology and the Eurostat case-study

This section provides a very brief summary of the most salient points of the previously submitted Deliverable D5.2. Readers already familiar with Deliverable D5.2 will find the material here familiar.

Based on an analysis of the requirements set forth by the automated journalism domain, chiefly an extremely high requirement for accuracy of the output, we interpreted that the needs of WP5 (and automated journalism in general) are best served by a natural language generation approach that is modular and at least partially rule-based, but also incorporates some neural processing, thus resulting in a hybrid approach. Importantly, we believe it is crucial that completely opaque 'black box' system are avoided.

Based on this analysis, we developed in Deliverable D2.4 a general EMBEDDIA text generation approach that is based on a pipeline of components with dedicated responsibilities. This structure allows for the individual components to be modified and replaced without affecting the rest of the pipeline. As the domain and language specific aspects of the pipeline are largely segregated (with the parts specific to both domain and languages further delegated to specific subcomponents), the system at large can be transferred to new domains and languages much more easily than applications based on a non-modular approach to NLG.

The architecture consists of eight primary stages: message generation (translating input data into atomic units of information that can be either included or omitted from the text), document planning (deciding what messages, and in which order, to include in the text), template selection (associating the messages with some linguistic content), lexicalization (deciding on the individual words to be used), aggregation (combining sentences where possible/necessary), named entity resolution (determining how to refer to domain entities, such as countries), morphological realization (inflecting words to their contextually correct forms) and surface realization (producing the final HTML-tagged text that can be shown to an end user). The modularity of the architecture allows us to employ both rule-based modules and neural (or otherwise machine learning based) modules in the same architecture. As the rule-based and machine learning approaches have complementing upsides and downsides, this hybrid approach allows us to always pick the option that fits best the requirements for any stage of the pipeline.

In the context of the Eurostat case study in Deliverable D5.2, we applied this text generation approach to a specific news generation problem. The Eurostat case study is a system implementation that produced textual news content from various data tables provided by Eurostat. The system was designed to be flexible with regard to addition and removal of data sources, but consequently is based on fundamental assumption about the format in which the data is provided and thus requires a degree of data preprocessing. Additionally, it is tailored towards a certain *type* of data and would not be suitable to, for example, reporting about a sports event. The work described in this deliverable is conducted in the context of the Eurostat system.

The system identifies the most pertinent – newsworthy – information from the data tables it is provided with and reports them in natural language. As a consequence of this adaptability to the various data tables, the system does not provide for any significant imputation or derivation of additional information beyond what is provided directly by Eurostat. The goal is to provide a starting point – a story 'blank' of relatively raw text material – which the journalist can refine to a larger story or focus down to a more detailed report.

An overview of the Eurostat system is shown in Figure 1. The central column of Figure 1 contains the main components of the architecture. These components, in conducting their processing, refer to various resources described in the right-most column of boxes. The resources are either dependent on the domain alone (light blue), dependent on the output language alone (red), or dependent on both (hatched). The system is interfaced with via the *API and Control* element, which provides an HTTP API for communication and also initiates the generation pipeline.

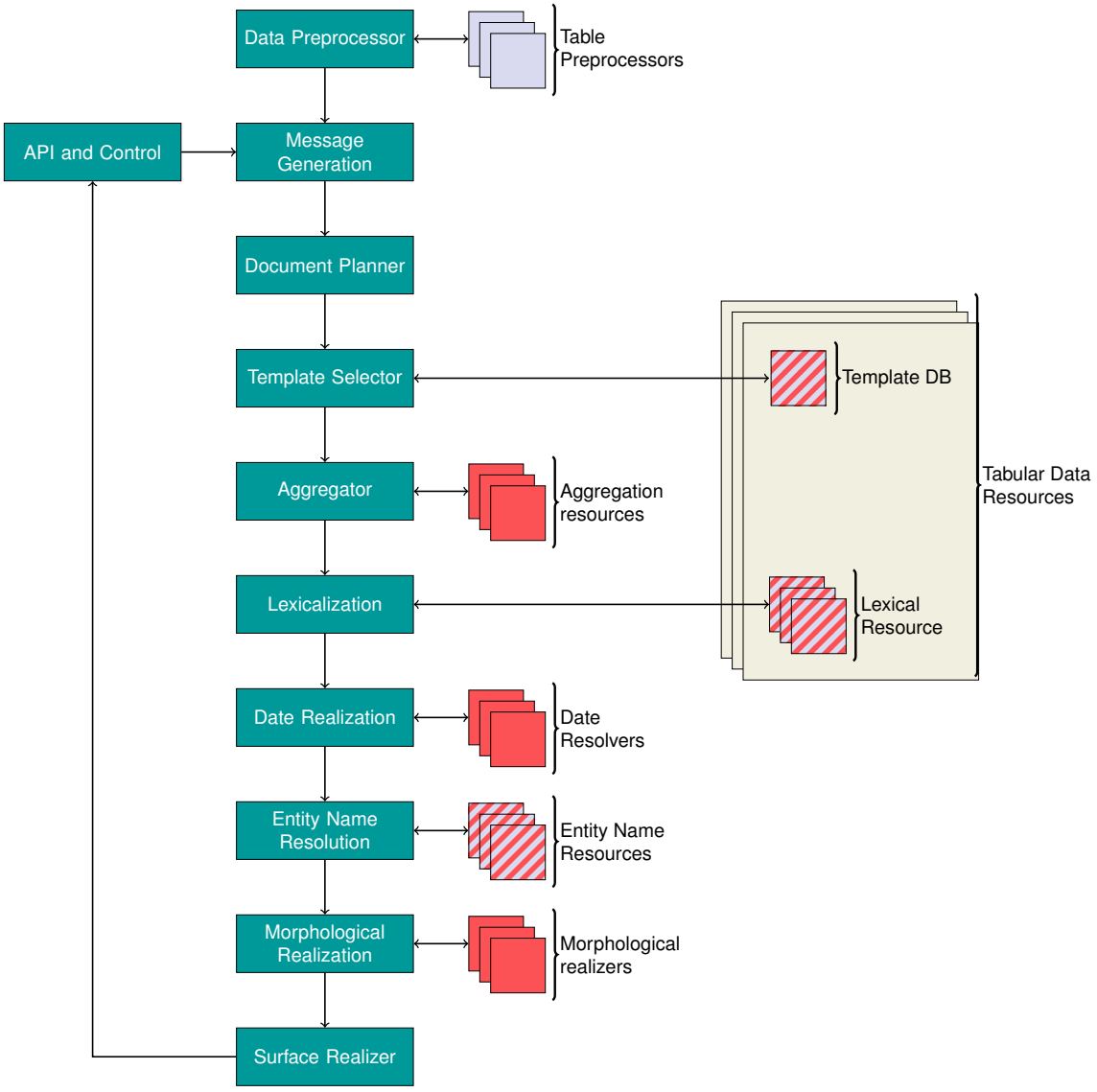

Figure 1: A high-level view of the Eurostat case study. On the two right-most columns, red boxes indicate modular components that are specific to a language, where as light-blue boxes indicate modular components that are specific to a domain. Hatched boxes are specific to both. Template databases and lexical resources are grouped into a set of Tabular Data Resources, each specific to a certain data table and language.

3 Extended language support for Eurostat news generation

We have significantly extended the number of languages supported by the Eurostat case study, by introducing support for Russian, Estonian and Slovene in addition to the previously supported Finnish, English and Croatian. This has entailed the creation of new language-specific resources for all of these languages, as well as minor modification to some pre-existing components.

In Deliverable D5.2 we discussed how the EMBEDDIA news generation approach is modular with respect to additions of new datasets and analysis. One other important aspect in the news generation domain is to be able to extend the system to new languages. Adding new languages to an NLG system is a non-trivial procedure: resources need to be provided for aggregation, date and entity name realization as well as morphological realization.

During the process of adding new languages, we observed that the human communication delays incurred especially during the translation of the templates (fundamental expressions used to express the information content, see Deliverable D5.2); these translations were a significant contributor to the time taken by the translation efforts. As these translations are the main resource needed when extending support to new datasets for languages already supported, we decided to further investigate the use of machine translation for producing draft translations of template files.

4 Neural network-based aid for adding languages

This section illustrates the solution we developed to reduce the burden of adding new languages to the EMBEDDIA news generation system. The solution makes use of neural Machine Translation (MT) models, which are discussed in Section 4.1. In Section 4.2, we show how multilingual embeddings can be used as a backbone to train MT models. These models are not used to translate the output of the NLG pipeline; instead, we use them as engines to support humans when adding new languages to the news generation pipeline. A detailed explanation of this approach is provided in Section 4.3.

4.1 Machine Translation

Machine Translation is a branch of natural language processing. Given some text in a *source* language, the goal of MT is to generate natural language in another language (*target*) preserving the meaning of the original text. According to the definition given by Reiter and Dale (1997, p. 57), MT does not fall within the scope of NLG, even if the output of MT systems is natural language, as the definition of NLG includes only systems that can produce text based on non-linguistic inputs.

Early MT approaches were based on handcrafted rules curated by linguistic experts. However, due to the inherent complexity of languages, it is problematic to model language irregularities in rule-based systems. Statistical machine translation (Brown et al., 1990; Koehn, Och, & Marcu, 2003) represents the first tentative step in the direction of automatically deriving translation rules from data, by using systems that can learn latent structures from parallel corpora, but this approach fails in learning the long-term dependencies that are present in the training data. Since the advent of deep neural networks and continuous representations, thanks to their capability in modelling language, end-to-end neural MT models have replaced rule-based and statistical methods in most of the automatic translation systems.

Neural MT systems make use of a single sequence-to-sequence neural network which directly transforms the source text – encoded in a continuous space – into the translated text. This is in contrast with statistical MT, which relies mainly on count-based models or incorporates neural models only as component of traditional statistical methods (see e.g., Schwenk, Dechelotte, & Gauvain, 2006; Zamora-Martínez, Bleda, & Schwenk, 2010; Schwenk, 2012; Devlin et al., 2014; Le, Allauzen, & Yvon, 2012;

Kalchbrenner & Blunsom, 2013; Cho et al., 2014).

The encoder-decoder architecture is the most commonly used neural MT architecture. It consists of an encoder network which computes a continuous representation of the source sequence, and a decoder network which generates the output sequence conditioned on the representation returned by the encoder. The training is performed in an end-to-end fashion and, as such, encoder-decoder models are regarded as comprising a single network.

4.2 Leveraging multilingual embedding models for MT

It has been observed that neural models making use of pre-trained weights can outperform models that are trained from scratch (see e.g., Devlin, Chang, Lee, & Toutanova, 2019, and references therein). In general, using pre-trained models allows for good performance with relatively short training time. Pre-training consists of training a model for a task, e.g., language modeling, and the use of the resulting network (or some layers of it) as a component in a larger neural model, to be further trained for another task, called *downstream task* (the optimization of the downstream task objective is called *fine-tuning*). Here, we show how pre-trained contextualized word embeddings can be fine-tuned in order to obtain encoder-decoder models for MT.

It is important to note that our purpose is not to provide a comprehensive set of MT models nor to compete with other neural MT architectures. Instead, we aim to define a training procedure that can be in principle applied to any language pair for which pre-trained models and a corpus of parallel data are available, and to streamline the development of MT models by taking advantage of pre-trained weights. Indeed, a huge number of pre-trained models are publicly available and several frameworks can be used to build the encoder-decoder architecture needed for the translation task. Parallel corpora that are necessary to fine-tune the models can be found for a large number of language pairs, see e.g., the OPUS corpus (Tiedemann, 2012).

For our implementation, we use the HuggingFace Transformers library, which allows easy access to a large number of pre-trained models and an easy way to combine them to construct the encoder-decoder architecture.[1] The architecture is based on the work of Rothe, Narayan, and Severyn (2020), where checkpoints of pre-trained transformer models are used to initialize both the encoder and the decoder of the sequence-to-sequence model. Thanks to the flexibility of the HuggingFace API, the same procedure works in general with other transformer-based pre-trained models (Rothe et al., 2020).

In our experiment, we considered the task of translating from Finnish to Estonian, and leverage the FinEst Bert model (Ulčar & Robnik-Šikonja, 2020), developed and pre-trained in WP1 and shown to outperform standard multilingual models in a range of tasks, to warm-start both the encoder and the decoder of a sequence-to-sequence model. We fine-tuned the model using the Finnish-Estonian parallel corpus provided by the Tatoeba challenge (Tiedemann, 2020).

As mentioned above, our goal is to show how publicly available pre-trained models can be used to develop MT models – and not competing with state of the art results – so we do not compare the performance of our Finnish-Estonian model with other implementations. Though, we believe that warm-starting an encoder-decoder model with pre-trained weights could improve the translation performance, especially when dealing with languages for which the amount of non-parallel data (necessary to pre-train the multilingual word embedding models) is relatively high, but parallel data (necessary for the fine-tuning) is scarce. This strategy could also be convenient when the computing power available to train the model is limited, as models initialized with pre-trained weights can achieve state of the art performance in downstream tasks with relatively short training times.

In the next section, we illustrate how to apply MT models to extend the EMBEDDIA news generation technology to new languages.

[1] `https://huggingface.co/transformers/`

4.3 Translating the language-specific resources of the EMBEDDIA news generation technology

In previous Deliverable D5.2, we described how the journalistic context of news generation sets requirements for transparency; accuracy; modifiability and transferability; fluency; data availability; and topicality. We identified that of these requirements, accuracy is especially crucial in that a system producing factually incorrect texts is unacceptable, irrespective of any other redeeming attributes it might have.

In order to add languages to our news generation system, a possible solution would be to feed the outputs of the NLG pipeline to an end-to-end MT program, obtaining the same news in the target language. While this would be a very *straightforward* implementation – and would probably work nicely most of the time in for languages that have sufficient amounts of parallel data to train accurate NMT models – this approach is not, in our view, suitable for news generation. For one, as the NMT model is given free reign to control every aspect of the output text, this effectively turns the whole system as a whole into a black box, a property we are very hard trying to avoid. Such black boxes would not be amenable to surgical corrections of the output and present unknown quality floors. Neural systems that generate language can also present subtle, but important, biases in their behaviours as discussed by Leppänen, Tuulonen, and Sirén-Heikel (2020), Ciora, Iren, and Alikhani (2021) and many others. As such, the strict legal and ethical responsibility guidelines of the journalism field would likely require every piece translated by the MT system to be inspected by a human before publication, thus severely limiting the applicability of the method.

As such, in our quest for automation, we do not apply MT models to the output of the pipeline, but rather leverage these models to draft the implementation of some of the pipeline components for new languages. In this way, instead of checking every single news piece in the target language, the human agent has to check and improve the pipeline components for the new language (which needs to be done only once during the system expansion), without having to check every output news story.

We obtain the pipeline components for a new language by translating the textual building blocks of each component from a language that is already supported by the system. For some of the components, such as date realization and entity name resolution, this is straightforward: the language-specific resources consist of dictionaries of terms, each corresponding to a possible value to be replaced in template slots, and easily translated by using the MT model. In any case, human intervention is needed to check and modify when necessary the results of the translation. Moreover, this method cannot be applied to all components of the NLG pipeline, as language-specific components are not always transferable across any language pair. For instance, the Aggregator component (see Deliverable D5.2) can be translated when the target and source languages have comparable syntactic rules. Even if this is the case, human intervention is most likely required to tweak the aggregation rules obtained from the translation. Another example is morphology realization, which could be transferred only across languages with equivalent morphosyntactic features. Due to its inherent complexity, we do not attempt to transfer morphology generation rules from source to target language, so that the implementation of morphology realization for the additional language is left to the user.

We first developed our approach using English as source language and Italian as target, followed by experiments translating Finnish to Estonian. We use here the English-Italian case to sketch our work, and report some examples for the Finnish to Estonian case. In order to add the Italian language resources to the system, from the English resources, we used the HuggingFace version of the Opus model `opus-mt-en-it`.[2] To add the Estonian resources from the Finnish pipeline, we used our MT model, presented in Section 4.1, and the Opus model `opus-mt-fi-et`.[3]

In Figure 2, we show the country name resources used in the Eurostat case study for English, and the equivalent resource translated into Italian.

[2] `https://huggingface.co/Helsinki-NLP/opus-mt-en-it`
[3] `https://huggingface.co/Helsinki-NLP/opus-mt-fi-et`

```
ENGLISH = {
    "AT": "Austria",
    "BA": "Bosnia and Herzegovina",
    "BE": "Belgium",
    "BG": "Bulgaria",
    "CZ": "Czechia",
    "DK": "Denmark",
    "DE": "Germany",
    ...
}
```

```
ITALIAN = {
    "AT": "Austria",
    "BA": "Bosnia Erzegovina",
    "BE": "Belgio",
    "BG": "Bulgaria",
    "CZ": "Cechia",
    "DK": "Danimarca",
    "DE": "Germania",
    ...
}
```

Figure 2: Country name resources for the EuroStat case study. On the left side, the resources used for English language; on the right side, the same resource obtained for Italian news generation obtained by using the English-Italian Opus model (Tiedemann, 2012).

Template (source language)

```
In {location}, the {value_type} was {value} {unit}.
```

Lexicalised Template

```
In Austria, the monthly growth rate of the harmonized consumer price
index for the category 'education' was 2 percentage points.
```

Realised slot values

```
{
    "location" -> "Austria",
    "value_type" -> "monthly growth rate of the harmonized consumer
                     price index for the category 'education'",
    "value" -> "2",
    "unit" -> "percentage points"
}
```

Realised slot values translated to Italian (from MT model)

```
{
    "location" -> "Austria",
    "value_type" -> 'tasso di crescita mensile dell'indice armonizzato
                     dei prezzi al consumo per la categoria "istruzione"',
    "value" -> "2",
    "unit" -> "punti percentuali"
}
```

Lexicalised Template Translated to Italian (from MT model)

```
In Austria, il tasso di crescita mensile dell'indice armonizzato dei
prezzi al consumo per la categoria "istruzione" è stato di 2 punti
percentuali.
```

Template (target language)

```
In {location}, il {value_type} è stato di {value} {unit}.
```

Figure 3: Sketch of the procedure adopted for template translation. The lexicalised template and the Fact values in the source language are translated to the target language, then the translated values are matched in the translated lexicalised template. In this example, colours are used to highlight the matching done to restore the slots in the translated template.

To get templates for a new language, it is not enough to feed the templates in the source language to a MT model. Indeed, as shown in Deliverable D5.2, templates are not natural language but contain slots that must be preserved in the translation. In order to get templates for the target language, we proceed as follows (see Figure 3 for a practical example, where an English template is translated to Italian).

For each template, a certain number of relevant Facts are picked from the data. Then, for each Fact:

1. The template is lexicalised by replacing in the slots the values of the considered Fact. Date realization, entity name resolution and morphological realization is performed over the lexicalised template. This results in natural language, reporting the Fact, in the source language (in the example of Figure 3, this is the *Lexicalised Template*).
2. Output from the previous step is translated to the target language by using a neural MT model

```
en: in {time}, in {location}, the {value_type}
was {value, abs} {unit} less than in US

it: nel {time}, in {location}, il {value_type} è stato di
{value, abs} {unit} inferiore a quello degli Stati Uniti
```

Figure 4: Example of template translation from English to Italian.

```
fi: {location, case=ssa}, {value_type} oli {value, abs}
{unit} vähemmän kuin Yhdysvalloissa

et: {location, case=ssa} oli {value_type} {value, abs}
{unit} vähem kui USAs
```

Figure 5: Example of template translation from Finnish to Estonian.

(see *Lexicalised Template Translated to Italian* in Figure 3).

3. The Fact values (after date realization and entity name resolution) are translated by using the neural MT model, in a similar way as it is done for the entity name resources (shown in Figure 2). In the example of Figure 3, this step consists in translating the *Realised slot values* to obtain the *Realised slot values translated to Italian*.
4. If all translated slots values from step 3 can be matched in the translated lexicalised template obtained in step 2, they are replaced by the corresponding slots obtaining the template for the target language. If there is no match, a new Fact is picked for the considered template, and the procedure is repeated. The matching procedure, in the example of Figure 3, is depicted by using colors to highlight the matching slots values.

In this way, with a sufficient number of Facts, it is usually possible to translate the templates that are required for a certain dataset. The number of facts required to successfully translate a template depends on the syntax similarity between source and target language. In general, the matching performed in step 4 above will be more likely when the languages are more similar and, in some cases, templates are not translated.

We also considered a "variational approach" for template translation. Within this approach, two realizations of the same template, differing only by one Fact value, are translated to the target language and compared. Assuming that the structure of the translated sentences does not depend on the specific value, the position of the slot can be inferred from the differences in the two translations. However, it is entirely possible that the same template gets very different translations when it is realised with different Fact values, and there is no way to detect this problem without introducing ad-hoc (and most likely language-dependent) heuristics. Our method for translating templates minimizes false positives in practical applications.

In Figure 4, we give an example of template translation, where the source language is English and the target language is Italian. Similarly, Figure 5 shows a template translated from Finnish to Estonian, by using the the encoder-decoder model based on FinEst Bert (trained as explained in Section 4.2).

We illustrate in Figure 6 the output of the system for a news story in Italian language, obtained from the EuroStat Italian pipeline and automatically transferred from English language, and in Figure 7 a similar example for Finnish and Estonian. Thanks to the relatively simple sentences and morphology of the target language, the output turns out to have only minor defects. Recall that morphology realization is

In January 2020, in Austria, the monthly growth rate of the harmonized consumer price index for the category 'education' was 0 points. [...] In Estonia, it was 1.3 percentage points more than in US. [...] Austria had the 4th highest monthly growth rate of the harmonized consumer price index for the category 'industrial goods' across the observed countries.

Nel Gennaio 2020, in Austria, il tasso di crescita mensile dell~~a~~' Indice armonizzato dei prezzi al consumo per la categoria "istruzione" è stato di 0 punti. [...] In Estonia, era 1.3 punti percentuali in più rispetto agli Stati Uniti. [...] L~~a~~' Austria ha registrato il 4° tasso di crescita mensile dell~~a~~' Indice armonizzato dei prezzi al consumo per la categoria "beni industriali" nei paesi osservati.

Figure 6: Example of output where source language is English, target language set to Italian. On the left, the same text obtained for English language. Red and green colors in the Italian is used to indicate errors identified by a native Italian speaker.

Itävallassa, kuukausittainen kasvu kuluttajahintaindeksissä 'koulutus' oli 0 yksikköä tammikuussa 2020. [...] Virossa, se oli 1.3 prosenttiyksikköä enemmän kuin Yhdysvalloissa. [...] Itävallan kuukausittainen kasvu kuluttajahintaindeksissä 'teollisuuden tavarat' oli 4. korkein.

Austria, igakuine kasv tarbijahinnaindeksis "koolitus" oli 0 ühikut jaanuar 2020. [...] Eesti oli see 1.3 protsendipunkti rohkem kui USAs. [...] Austria igakuine kasv tarbijahinnaindeksis "tööstuskaubad" oli 4 kõrgeim.

Figure 7: Example of output where source language is Finnish, target language set to Estonian. On the left, the same text obtained for Finnish language.

out of the scope of this report; morphological realizers exist for the EMBEDDIA languages and will need to be added manually.

In the Italian news story (Figure 6), red strike-through and green underlined characters denote, respectively, characters that should be removed and inserted in order to have a grammatically correct output. In this specific case, all errors in the output derive from wrong article forms. The partitive article *della* (in English *of the*) is wrongly used in its feminine form in front of the masculine noun *Indice* (*Index*). This comes from the fact that the template used to realise the sentence was translated after replacing a value with a feminine noun in the slot, so that the feminine article form ended up in the template.

To recap, our strategy is to start from a source language that is already supported by the NLG system and obtain a *draft implementation* of the NLG pipeline for a target language, excluding morphology generation. The amount of modifications that need to be performed by the user in the resulting pipeline will depend on the syntactic similarity between the source and target languages.

We have implemented the translation model as a command line interface tool for the EuroStat case study. The tool allows the user to obtain a draft translation of the template string in a form where it can be inserted directly into a suitable tabular data resource (see Deliverable D5.2). An abridged example of using the tool to translate a template file from English to Italian is shown in Figure 8. The example shows how the tool is able to reconstruct the structure of the template file, including the template rules.

5 Evaluation

A formal evaluation of the NLG systems developed in Task 5.1 will be conducted within the scope of Task 5.4. At the same time, even our initial successes in implementing the systems and producing meaningful textual output points towards at least a reasonably successful NLG approach.

Our experiments so far indicate that the machine translation approach is useful for producing draft translations of the template files. The approach shortens the translation time by allowing non-native developers to produce the drafts and let native speakers only check and correct the results. This is in strong contrast to the manual alternative where a native speaker either translates both complete lexicalized sentences (as in the MT approach) or needs to get familiar with the template format and translate them.

6 Conclusions and further work

In the previous deliverable D5.2, we described two case studies exemplifying our approach for adapting the multilingual language generation technology developed in Task T2.3 to automated journalism. They highlighted the adaptability of the underlying technology with regard to both *datasets* (EuroStat case study) and *analyses* within a dataset (COVID-19 case study).

The current deliverable improves on that previous work with respect to *languages*. First, we have doubled

```
(.venv) $ python eunlg/template_translator.py --dataset cphi --
source_language en --target_language it --model Helsinki-NLP/opus-
mt-en-it
 [...]
 TRANSLATED TEMPLATES:
 ---------------
 it: nel {time}, in {location}, il {value_type} è stato di {value} {unit}
 it: nel {time}, il {value_type} è stato di {value} {unit}
 it: in {location}, il {value_type} è stato di {value} {unit}
 it: Il {value_type} è stato di {value} {unit}
 it: nel {time}, in {location}, era {value} {unit}
 it: nel {time}, era {value} {unit}
 it: in {location}, era {value} {unit}
 it: era {value} {unit}
 it: in {location}, nel {time}, il {value_type} è stato di {value} {unit}
 | value_type = cphi:.*, value_type != .*:rank.*, value_type != .*:comp_.*

 it: nel {time}, in {location}, il {value_type} è stato di {value} {unit} superiore alla media UE
 it: nel {time}, il {value_type} è stato di {value} {unit} superiore alla media UE
 it: in {location}, il {value_type} è stato di {value} {unit} superiore alla media UE
 it: Il {value_type} è stato di {value} {unit} superiore alla media UE
 it: nel {time}, in {location}, è stato di {value} {unit} in più rispetto alla media UE
 it: nel {time}, è stato di {value} {unit} in più rispetto alla media UE
 it: in {location}, è stato di {value} {unit} in più rispetto alla media UE
 it: è stato di {value} {unit} in più rispetto alla media dell' UE
 it: in {location}, nel {time}, il {value_type} è stato di {value} {unit} rispetto alla media UE
 | value_type = cphi:.*:comp_eu, value_type != .*:rank.*, value > 0

 [...]

 it: L' {location} aveva il {value} ° {value_type} nei paesi osservati.
 it: nel {time}, l' {location} aveva il {value} ° valore più basso per i paesi osservati
 it: L' {location} aveva il {value} ° valore più basso per i paesi osservati
 it: nel {time}, {location} {value} ° {value_type} più basso
 | value_type = cphi:.*:rank_reverse.*

 ---------------
```

Figure 8: An abridged example of the output obtained when the machine translation tool is used to translate a template file from English to Italian.

the number of languages supported by the Eurostat system from three to six. Second, motivated by the aforementioned work, we described a method for machine translation of significant portions of the language- and domain-specific assets within the system. Developed into a practical command line tool, this translation method helps develop the first drafts of language- and domain-specific assets, thus shortening development times. This approach allows for a human to remain in control, helping to meet the requirements of automated journalism such as correctness and transparency and avoiding problems associated with black box systems.

We intend to continue working on and fine-tuning the systems to the end of the project based on feedback obtained from the domain expert project partners, as well as the results obtained through the evaluation of the systems as conducted within the scope of Task T5.4.

7 Associated outputs

The work described in this deliverable has been conducted within the context of the earlier iteration of the Eurostat system described in Deliverable D5.2 of this same task. A such, the contributions described above have also been incorporated into that same source code repository:

Description	URL	Availability
EuroStat news generation system (source code)	`https://github.com/EMBEDDIA/eurostat-nlg`	To become public

We are currently preparing scientific publications based on the works described both here and in Deliverable D5.5, and as such the source code repository listed below is not yet public. The source code will be made public later with a suitable open source license, as the code bases stabilizes and the scientific publications being prepared become public. As the morphological models used in the systems are licensed under the GPL license, we expect that the system themselves will also be released as under the GPL license.

References

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., ... Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79–85. Retrieved from `https://aclanthology.org/J90-2002`

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1724–1734).

Ciora, C., Iren, N., & Alikhani, M. (2021). Examining covert gender bias: A case study in turkish and english machine translation models. *arXiv preprint arXiv:2108.10379*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Naacl*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., & Makhoul, J. (2014, June). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1370–1380). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P14-1129` doi: 10.3115/v1/P14-1129

Kalchbrenner, N., & Blunsom, P. (2013, October). Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1700–

1709). Seattle, Washington, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D13-1176

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 human language technology conference of the north American chapter of the association for computational linguistics* (pp. 127–133). Retrieved from https://aclanthology.org/N03-1017

Le, H. S., Allauzen, A., & Yvon, F. (2012, June). Continuous space translation models with neural networks. In *Proceedings of the 2012 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 39–48). Montréal, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N12-1005

Leppänen, L., Tuulonen, H., & Sirén-Heikel, S. (2020). Automated journalism as a source of and a diagnostic device for bias in reporting. *Media and Communication*.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87.

Rothe, S., Narayan, S., & Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, *8*, 264–280. Retrieved from https://aclanthology.org/2020.tacl-1.18

Schwenk, H. (2012, December). Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012: Posters* (pp. 1071–1080). Mumbai, India: The COLING 2012 Organizing Committee. Retrieved from https://aclanthology.org/C12-2104

Schwenk, H., Dechelotte, D., & Gauvain, J.-L. (2006, July). Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 main conference poster sessions* (pp. 723–730). Sydney, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P06-2093

Tiedemann, J. (2012, May). Parallel data, tools and interfaces in OPUS. In *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 2214–2218). Istanbul, Turkey: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf

Tiedemann, J. (2020, November). The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the fifth conference on machine translation* (pp. 1174–1182). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.wmt-1.139

Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In P. Sojka, I. Kopeček, K. Pala, & A. Horák (Eds.), *Text, speech, and dialogue TSD 2020* (Vol. 12284). Springer. Retrieved from https://doi.org/10.1007/978-3-030-58323-1_11

Zamora-Martínez, F., Bleda, M. J., & Schwenk, H. (2010). N-gram-based machine translation enhanced with neural networks for the french-english btec-iwslt'10 task. In *Iwslt.*