Research and Innovation Action (RIA) H2020 – 957017

Stream Learning for Multilingual Knowledge Transfer

https://selma-project.eu/

D2.5 Intermediate release of stream learning and entity linking capabilities

Work Package	2
Responsible Partner	Priberam
Author(s)	João Figueira
Contributors	Afonso Mendes, Diogo Pernes
Reviewer	Salima Mdhaffar
Version	V1.0
Contractual Date	31 December 2022
Delivery Date	22 December 2022
Dissemination Level	Public

Version History

Version	Date	Description
0.1	03/11/2022	Introduction and Initial Table of Contents (ToC)
0.2	11/12/2022	Input
0.3	21/12/2022	Internal Review
1.0	22/12/2022	Publishable version

Executive Summary

This intermediate deliverable cumulatively describes the release of software components for stream learning and entity linking developed within WP2 and their integration with the SELMA orchestration platform. These components are mostly applicable to UC1, the Media Monitoring Platform. Major improvements on language support by use of transfer learning have been achieved with this second batch of components.

SELMA's approach to named entities, a major challenge in transcription, is to continuously learn new named entities from the reference stream and link them to a knowledge base (e.g., Wikipedia)

This document provides an overview of the current release of the stream learning and entity linking capabilities to be followed by the final releases later in the project.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

This intermediate deliverable cumulatively describes the release of software components for stream learning and entity linking developed within WP2 and the integration with the SELMA orchestration platform. These components are mostly applicable to UC1, the Media Monitoring Platform. We address two components of the Natural Language Processing (NLP) document enrichment pipeline, namely, Named Entity Recognition (NER), and Multilingual Named Entity Linking (NEL). The objective of this report is to describe those components from the point of view of the release of software components. This deliverable should be read in conjunction with D2.4, where a full technical description is reported.

All components are deployed as Docker containers (www.docker.com) and will be made available at https://hub.docker.com/orgs/selmaproject, expose REST APIs and provide swagger documentation pages (https://swagger.io/). These components are integrated with the SELMA orchestration platform and are already being used by Use Case 1 (UC1) (https://app.monitio.com).

2. Released components

2.1 Annotation tool

An annotation tool for Named Entity Recognition was developed by Priberam and deployed at https://www.priberam.com/annotate. It is being used by Priberam, IMCS, and DW for the development of a multilingual named entity dataset available in several languages. Priberam has already annotated news documents in Portuguese (3000 documents), French (3000 documents), Spanish (in progress), German (3000 documents), and English (6000 documents). IMCS is currently annotating Latvian (in progress) and is planning to start Russian. DW already had the necessary training sessions with the Priberam linguists' team and will soon start annotating an Arabic dataset.

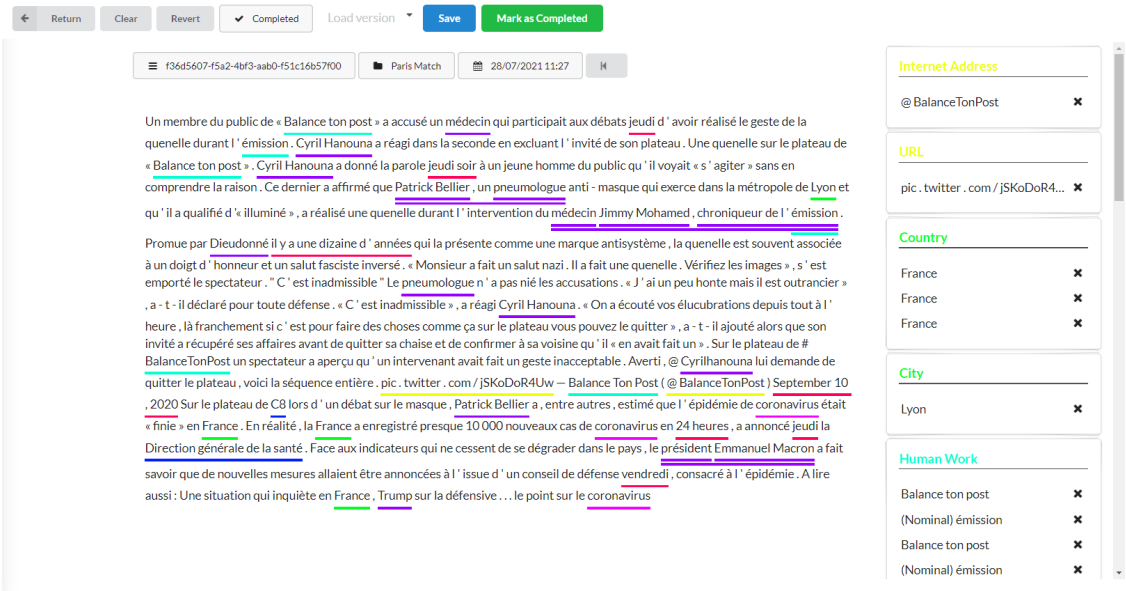

Figure 1 *NER Annotation tool*

2.2 Named Entity Recognition models using HNNNER

During the reporting period, we deployed a set of trained models using the datasets developed in the scope of the SELMA project according to the annotation guidelines defined in D6.1. We deployed, in particular, Spanish, English, Portuguese, German and French and we are

evaluating ether the model trained on the already available Latvian dataset already achieves the minimum requirements to be deployed on UC1.

As described in D2.1, we developed two different models, the stack-LSTM and the Biaffine model for Hierarchical Nested Named Entity Recognition. As reported in D2.1, both models presented similar F1 scores in our news test Datasets, where the stack-LSTM has shown better performance characteristics to be deployed in the UC1 scenario.

These models have been deployed as a docker container exposing a REST API which is then used by the SELMA orchestration platform. The docker containers were deployed in machines with GPUs, so that the document processing through-output was compatible with UC1 requirements. The complete NLP pipeline used on UC1 is currently being able to ingest and process 150 000 documents per day.

The NER rest API is composed by a single method that receives a json object with the text as in the following example:

```
POST /predict/ HTTP/1.1
Host: pbacomp03.interno.priberam.pt:8800
Content-Type: text/plain
Content-Length: 202
{
  "text": "Susan Kersch-Kibler, Gründerin der Agentur Delivering Dreams, hat ihre Leihmütter kurzerhand ins Ausland verfrachtet – nur um sie zum Geburtstermin wieder in die Ukraine zu befördern. "
}
```

And produces the following output where for each entity a json object is returned according to the following definition:

mention_id	Sequential id
mention	Mention text
total_offset	Offset from the beginning of input text
end_offset	End offset from the beginning of text
sentence_offset	Near context offset form the beginning of text
near_context	Text sentence where the mention was detected

ner_tag	NER type as defined in the Ontology
ner_type	NAM for named entity or NOM for nominal entity
ner_modifiers	Modifiers applied to the NER type like collective, negated etc. (see ontology in D6.1)


```
[
    {
        "mention_id": 0,
        "mention": "Susan Kersch-Kibler",
        "length": 19,
        "total_offset": 0,
        "end_offset": 19,
        "sentence_offset": 0,
        "near_context": "Susan Kersch-Kibler, Gründerin der Agentur
Delivering Dreams, hat ihre Leihmütter kurzerhand ins Ausland verfrachtet –
nur um sie zum Geburtstermin wieder in die Ukraine zu befördern. ",
        "sentence_id": 0,
        "ner_tag": "people",
        "ner_type": "NAM",
        "ner_modifiers": []
    },
    {
        "mention_id": 1,
        "mention": "Delivering Dreams",
        "length": 17,
        "total_offset": 43,
        "end_offset": 60,
        "sentence_offset": 0,
        "near_context": "Susan Kersch-Kibler, Gründerin der Agentur
Delivering Dreams, hat ihre Leihmütter kurzerhand ins Ausland verfrachtet –
nur um sie zum Geburtstermin wieder in die Ukraine zu befördern. ",
        "sentence_id": 0,
        "ner_tag": "organization->commercial_company",
        "ner_type": "NAM",
        "ner_modifiers": []
    },
    {
        "mention_id": 2,
        "mention": "Agentur Delivering Dreams",
        "length": 25,
        "total_offset": 35,
        "end_offset": 60,
```

```
        "sentence_offset": 0,
        "near_context": "Susan Kersch-Kibler, Gründerin der Agentur
Delivering Dreams, hat ihre Leihmütter kurzerhand ins Ausland verfrachtet –
nur um sie zum Geburtstermin wieder in die Ukraine zu befördern. ",
        "sentence_id": 0,
        "ner_tag": "organization->commercial_company",
        "ner_type": "NAM",
        "ner_modifiers": []
    },
    …
```


Multilingual Named Entity Recognition

During the second reporting period we have deployed a new fully multilingual version of the NER component. The component supports 100 languages, the same as the base model xlm-roberta-base. The component was trained on English, German, Spanish, French, Latvian and Portuguese. It was evaluated on unseen languages during the training in Dutch, Ukrainian and Turkish a very good zero-shot performance. Qualitative evaluation leads us to believe that the component zero-shots with very reasonable results to many of the other languages. It is now integrated in UC1 and the users perception is very good across all languages represented in the platform. Evaluation for these component can be found in D2.4. Besides the language support, the model allows us to better scale the processing pipeline since we only use the resources (CPU, GPU, RAM) for one model instead of one per language.

The deployment details are the same as the previous NER component.

2.3 Multilingual Entity Linking with Wikidata/Wikipedia as the Knowledge Base

A first version of the Entity Linking models described in D2.1 were deployed for UC1 in the MONITIO platform. The model deployed currently supports the same languages as the NER models described on the previous section (Spanish, Portuguese, German and French) covering a total of 14 322 317 different entities. Since Wikidata and Wikipedia are constantly being updated and new entities inserted we have implemented an automatic procedure to collect and incrementally update the knowledge base data. This procedure incrementally trains and updates the representations for the entities as described in D1.2.

The model was deployed as docker container exposing a REST API. A single instance is able to process 2 documents per second on average when deployed over GPU. In order to cope with the current MONITIO stream, we have currently deployed two instances of the service.

The REST API exposes two methods: one to obtain the possible candidates for a given mention and another to perform the actual EL for a given document.

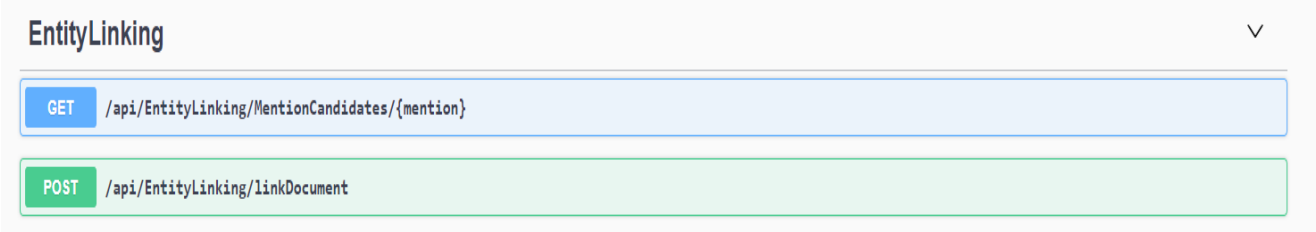

Figure 2 *Swagger method definition for the EL component*

For each linked entity, the service provides the Wikidata identifier for the entity and the Wikipedia title in the original language and English. In the next version of the service, we will include additional metadata regarding the entities like gender, date of birth, country, occupation, ethnic group, etc... as defined in the SELMA requirements for the diversity use case (see D1.1 - 3.1.1.2 Diversity).

Currently, the service returns for each entity (additional information is available in the swagger page):

baseForm	English Wikipedia title if available; if not, the same as currlangForm
currLangForm	Wikipedia title in the original language of the document
id	Wikidata unique identifier
type	one of people, location, gpe, organization and event

Sample Json output from the EL service:

```
{
    "entities": [
```

```
{
    "entity": {
        "baseForm": "Aung San Suu Kyi",
        "currlangForm": "Aung San Suu Kyi",
        "id": "Q36740",
        "lowest_confidence": 21.091115864011837,
        "type": "people"
    },
    "mentions": [
        {
            "confidence": 1,
            "endPosition": {
                "chunk": 0,
                "offset": 37
            },
            "ner_type": "people",
            "sourceDocument": null,
            "startPosition": {
                "chunk": 0,
                "offset": 21
            },
            "text": "Aung San Suu Kyi"
        },
        {
            "confidence": 1,
            "endPosition": {
                "chunk": 0,
                "offset": 296
            },
            "ner_type": "people",
            "sourceDocument": null,
            "startPosition": {
                "chunk": 0,
                "offset": 284
            },
            "text": "Aung Suu Kyi"
        },
….
```


Extended Multilingual Entity Linking

During the second reporting period, we deployed a new entity-linking component with support for 40 languages. The following table lists languages currently supported by the EL model. Technical Description for this module can be found on D2.4. The deployment details are the same as the previous version.

Language Code	Language
am	Amharic
ar	Arabic
bg	Bulgarian
bn	Bengali
bs	Bosnian
ca	Catalan
cs	Czech
de	German
el	Greek
en	English
es	Spanish
fa	Farsi
fi	Finnish
fr	French
ha	Hausa
he	Hebrew
hi	Hindi
hr	Croatian
hu	Hungarian
id	Indonesian
it	Italian
ja	Japanese
lv	Latvian

mk	Macedonian
nl	Dutch
no	Norwegian
pl	Polish
ps	Pashto
pt	Portuguese
ro	Romenian
ru	Russian
sq	Albanian
sr	Serbian
sv	Swedish
sw	Swahili
tr	Turkish
uk	Ukrainian
ur	Urdu
zh	Chinese

Table 1 *EL Language support*

NER/EL user feedback in the MONITIO platform

According to the requirements defined for UC1, the MONITIO platform was extended to gather NER / NEL corrections and additions.

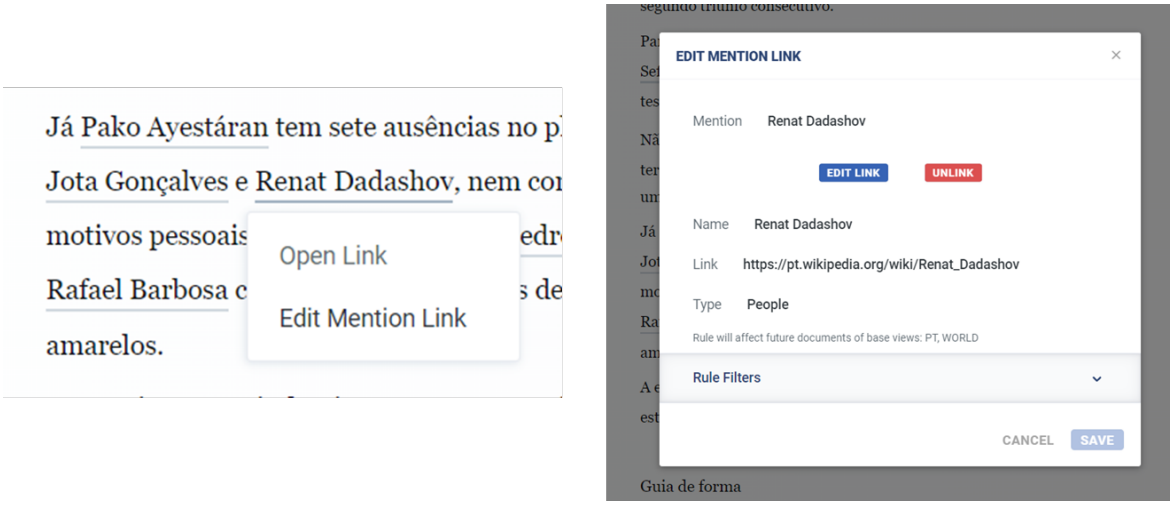

Figure 3 *Feedback collection user interface for entity linking and NER*

The functionality is already being used by test users and the data is being collected. Research on user feedback using this data is starting in the context of WP2 and will be integrated later.

2.4 Rule-Based Stream Learning for NEL (PiniTree Ontology Editor)

The SELMA partner IMCS, the University of Latvia, has been involved in the NEL topic for several years (Barzdins, 2020; Paikens, 2016a), jointly with the Latvian national news agency LETA and PiniTree.com startup. This has resulted in the development of the commercial PiniTree.com ontology editor with integrated rule-based Stream learning of Named Entity Linking aliases as part of the entity database, against which the Named Entities are being Linked. PiniTree editor is one of the tools being integrated into the SELMA Platform. In addition, the LETA use case (described in detail in deliverable D1.2) is available for wider exploitation along with other SELMA components. Within the initial release of the SELMA Platform, PiniTree is integrated into the Use Case 0 as the backend content management system accessed via “Publish” button.

PiniTree technically is a universal web server with integrated database and user management. PiniTree is distributed as a single precompiled binary file downloadable from https://pinitree.com, therefore no specific installation is required – in Linux PiniTree server is started by directly executing the PiniTree binary file from the command-line:

```
bash % ./pinitree.linux-amd64 -p 7777 -a
```

The rest of the PiniTree web server configuration takes place through the graphical web interface. The first time when trying to connect to the newly started PiniTree web server, it will display the following prompt to create the first (admin) user. Additional users with various privileges can be added later by logging in as admin.

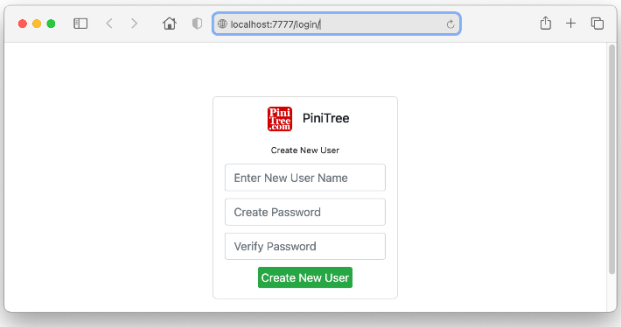

Figure 4 *Creating an admin user for the PiniTree ontology editor*

Out-of-the-box PiniTree can be used as a simple web server - click on the PiniTree.com logo in the upper left corner and upload few files through "AddFile" button. Then click on the "Open Uploads" to browse the uploaded files. If you would have uploaded also an "*index.html*" file, that would have been displayed instead of the file list – anyone with the correct URL now can access it.

The core use case for the PiniTree software is creating and maintaining a Named Entity Linked document store similar to WikiPedia, as described in detail in the previous sub-section.

PiniTree server will create a */data* folder in the current directory - this is the only place where PiniTree stores all its runtime data. For backup or cloning your PiniTree instance, back up or copy this */data* folder to another computer as needed. Alternatively, one can symlink */data* folder to another disk or directory.

Additional PiniTree executable options can be looked up with "*./pinitree.linux-amd64 -help*" command.

PiniTree ontology editor can be controlled not only from the graphical user interface via web browser, as described in Section 3.1, but also programmatically via REST API illustrated in

Figure 4. A distinctive approach in this REST API is the long-polling “wait” call, which allows external systems to react in real-time to the PiniTree database changes (e.g., due to user actions or other REST API calls) without placing any control-flow logic inside the PiniTree server itself. This allows a universal PiniTree server to orchestrate multiple parallel interactive control flows simultaneously.

Figure 5 *REST API access to PiniTree ontology editor database*

Via REST API PiniTree editor, the internal database can be integrated in real-time with the external data and processing sources and function as a component of a larger system. This is how PiniTree ontology editor is being integrated into the SELMA platform Use Case 0.

3. Future plans

This intermediate software release already covers tasks T2.1 and T2.2 of WP2. Entity linking already has broad language support and shows very good performance on the test datasets. We achieved very good language transfer results on the task of Named Entity Recognition.

For the next releases of software components, our focus will be:

1) Deploy new NER models as they new training data is being annotated;
2) Further increase the language cover for our EL models;
3) Joint entity-linking and NER models;
4) Release of EL using stream learning and clustering.

Bibliography

Barzdins, G., Gosko, D., Cerans, K., Barzdins, O. F., Znotins, A., Barzdins, P. F., Gruzitis, N., Grasmanis, M., Barzdins, J., Lavrinovics, I., Mayer, S. K., Students, I., Celms, E., Sprogis, A., Nespore-Berzkalne, G., Paikens, P. (2020b). Pini Language and PiniTree Ontology Editor: Annotation and Verbalisation for Atomised Journalism. *In: ESWC 2020 Satellite Events. LNCS, Volume 12124, pp. 32-38.*

Peteris Paikens; Guntis Barzdins; Afonso Mendes; Daniel Ferreira; Samuel Broscheit; Mariana S. C. Almeida; Sebastiao Miranda; David Nogueira; Pedro Balage; Martins, Andre F. T. (2016a). SUMMA at TAC Knowledge Base Population Task 2016, *DOI: 10.5281/zenodo.827317*

Znotiņš, Artūrs & Barzdins, Guntis. (2020). LVBERT: Transformer-Based Model for Latvian Language Understanding. *Baltic HLT, IOS Press, pp. 111-115, DOI 10.3233/FAIA200610.*

Znotins A, Cirule E. (2018). NLP-PIPE: Latvian NLP Tool Pipeline. *In: Human Language Technologies - The Baltic Perspective. vol. 307. IOS Press; 2018. p. 183–189.*

Paikens P. (2016b). Deep Neural Learning Approaches for Latvian Morphological Tagging. *In: Baltic HLT; 2016. p. 160–166.*

João Santos, Afonso Mendes and Sebastiao Miranda, “Simplifying News Clustering Through Projection from a Shared Multilingual Space” in Text2Story at ECIR, 2022.

You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H., & Zhu, S. (2019). AttentionXML: Label Tree-based Attention-aware Deep Model for High-performance Extreme Multi-label Text Classification. *Advances in Neural Information Processing Systems*.