D2.4

Cross-lingual Lexical Resource Linking Web Service (software)

Authors: Federico Martelli, Roberto Navigli, Paola Velardi, Atul Kumar Ojha, John P. McCrae

Date: 31 January 2022

H2020-INFRAIA-2016-2017

Grant Agreement No. 731015

ELEXIS - European Lexicographic Infrastructure

D2.3 CROSS-LINGUAL LEXICAL RESOURCE LINKING WEB SERVICE (SOFTWARE)

Deliverable number: 2.4

Dissemination Level: Public

Delivery Date: 31 January 2022

Version: Final

Authors: Federico Martelli, Roberto Navigli, Paola Velardi

Project Acronym:	ELEXIS
Project Full Title:	European Lexicographic Infrastructure
Grant Agreement No.:	731015

Deliverable/Document Information

Project Acronym:	ELEXIS
Project Full Title:	European Lexicographic Infrastructure
Grant Agreement No.:	731015

Document History

Version Date	Changes/Approval	Author(s)/Approved by
31 January 2022	-	-

Table of Contents

List of Tables

List of Figures

1 Introduction

This document describes the software infrastructure released as deliverable D2.4 "Cross-lingual Lexical Resource Linking Web Service (software)" related to task 2.3 "Cross-lingual mapping through shared conceptualization" - work package 2: JRA Interoperability and Linked (Open) Data.

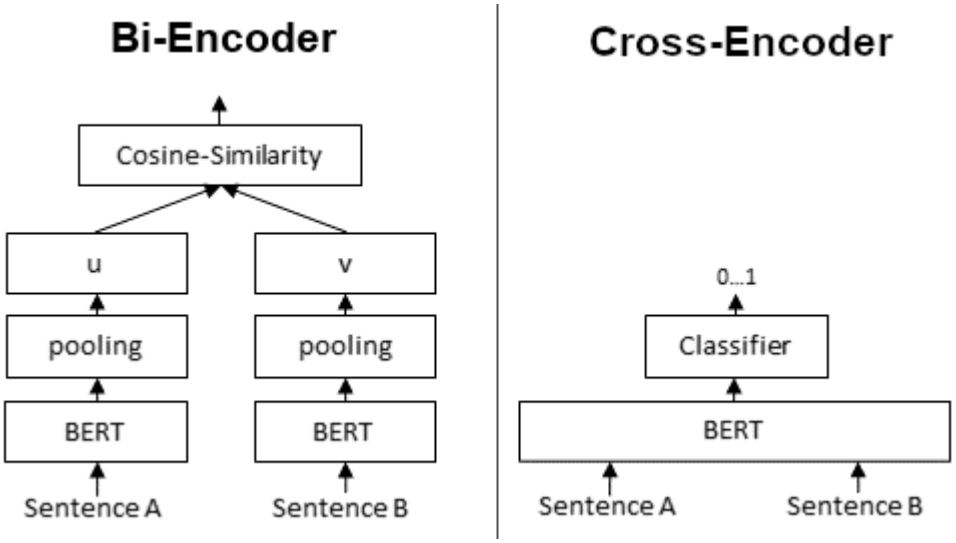
The main objective of this deliverable is the creation of a linking web service which produces a mapping between two dictionary definitions in a cross-lingual scenario. Specifically, the released service is capable of linking a definition derived from a dictionary provided within the ELEXIS Consortium and an English definition in the BabelNet semantic network [5]. Importantly, this linking process will make it possible to map the ELEXIS dictionaries at definition level by pivoting through BabelNet.

In what follows, we provide a task formulation and introduce the BabelNet-linker tool which allows such mapping to be achieved by relying on state-of-the-art Transformer-based neural architectures. Finally, we provide detailed documentation regarding the usage of our API. Our linking infrastructure is available at babelnet.linkingmachine.org. Documentation regarding its usage can be found at babelnet.linkingmachine.org/docs. Furthermore, we released our code to perform cross-lingual lexical resource linking at https://github.com/elexis-eu/BabelNet-linker.

2 BabelNet-linker

Let δ_λ be a definition in language λ derived from a dictionary provided within the ELEXIS Consortium and σ_ε a set of definitions in English derived from the BabelNet semantic network. Our task consists in identifying the definition(s) in σ_ε which share the same semantics with δ_λ. In what follows, we first detail the architectures which we designed to address this task. Subsequently, we describe the data which we used as well as the achieved results.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.



3

Fig. 1 Depiction of the embedding-based and cross-encoder-based architectures for cross-lingual dictionary linking.

2.1 Architectures

To address the cross-lingual lexical resource linking task, we design the following Transformer-based architectures:

- Embedding-based (Bi-encoder): This model is based on the embedding of two definitions provided as input separately and the cosine similarity between them is used as the similarity score.
- Cross-encoder-based: This model is based on the embedding of two definitions using a cross-encoder, which receives as input two definitions simultaneously, computing the similarity score with a Dense layer.

Furthermore, we experiment with a third architecture, namely:

- ESCHER-based architecture: This model is based on the work proposed by [3], in which WSD is framed as a Question Answering task.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

The embedding-based approach has the advantage of being able to compute the similarity score much faster, by encoding each definition once, i.e. given *N* source senses and *M* target senses for a given lemma and POS tag, it only needs to encode *N+M* definitions. However since it is based on independent embeddings, there is no interaction between the two definitions.

On the the hand, the cross-encoder requires more computation time, since it encodes both definitions simultaneously, i.e. provided *N* source definitions and *M* target definitions for a given lemma and POS tag, the cross-encoder encodes *N*M* inputs. However, such architecture is able to compute the similarity score between the two definitions as input, enabling the model to encode the shared information from both definitions, thus leading to a better performance.

Instead of BERT [4], we use multilingual encoders that have been pretrained on NLI tasks:

- mDeBERTa
- paraphrase-multilingual-mpnet-base-v2
- LaBSE
- xlm-roberta-large-xnli-anli

The first two architectures are trained on a binary task which aims at determining whether a given pair of definitions belong to the same synset or not. This configuration is different from the one adopted at testing time, since at training time the model is only given a pair and the loss is computed based on the similarity score between the two definitions in a binary fashion.

On the other hand, when evaluating the model we are given a single definition in a source language, and we link it to the corresponding definition(s) from BabelNet. Furthermore, we may have negative instances in which we do not have a definition in the target language and the model should give a low score to all the candidate definitions. Performance is therefore based on this setup, for which we have positive and negative examples.

To overcome the aforementioned mismatch between training and evaluation in terms of architecture, we explore the use of a recent state-of-the-art model for Word Sense Disambiguation (WSD) called ESCHER. In this work, the task of WSD is reframed as a Question Answering task, where the input sentence and a target word is the question and the possible definitions for that word is given as the

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

context, therefore the task consists in predicting the correct span for the correct definition. In our case, the source definition in a given language is provided as the question, and our candidate definitions from BabelNet represent our context. Furthermore, we simultaneously train a feed-forward classifier that takes the predicted span as input to classify whether there is a correct candidate definition, since we can have instances with no answer, i.e. no match in BabelNet. Thanks to this architecture, we are able to train and evaluate on the same task.

We test our architectures with the pre-trained Transformers listed above and select the best performing one for each one.

2.2 Data

In this section, we describe the data which we used to train and test our models.

As far as the training data is concerned, we first extract a list of approximately 27k ambiguous English words covering all four open-class parts of speech. For each English lemma *L* we extract all English synsets containing *L* and its definitions. Subsequently, for each extracted synset we obtain the lemma for its main sense in the set of languages covered by the task, *L_lan*. For each *L_lan* we extract all synsets containing *L_lan* in the same language, as well as its definitions in its language and English. For those in which definitions in both the original language and English are available, we obtain one positive match for training. Negative examples are obtained by matching a definition in the original language with any of the English definitions for other synsets containing *L_lan.* We also include the training data from the Monolingual Word Sense alignment task, in which definitions are paired within the same language, hence matches and candidates belong to the same language. For the binary training setup we generate a balanced amount of negative examples, while for ESCHER we use all the negative candidates. We create training datasets in the following languages: Basque, Bulgarian, Danish, Dutch, English, Estonian, French, German, Hausa, Hungarian, Irish, Italian, Portuguese, Russian, Slovak, Slovene and Spanish. Furthermore, we used additional training data derived from [6].

As far as testing data is concerned, we manually create a gold standard which we describe in the next section.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

2.2.1 Manually-curated gold standard

For each language mentioned in section 2, we produce a manual gold standard in the following way. First, we extract lemmas and definitions used for the shared task on Monolingual Word Sense alignment (MWSA). We query BabelNet using the lemmas from the task and designed an annotation setup in which expert annotators were given a definition and lemma *L* from a source language and a set of BabelNet definitions in English for synsets containing the lemma *L*. Subsequently, annotators selected which synsets match the source definition, or whether there is no match. The supported languages are: Bulgarian, Danish, Estonian, Hungarian, Irish, Italian, Portuguese, Slovenian and Spanish.

Lang	Nouns	Verbs	Adjectives	Adverb	Total
BG	180 (267)	13 (23)	28 (50)	1 (3)	222 (343)
ES	158 (490)	79 (183)	84 (220)	1 (4)	322 (897)
ET	90 (125)	15 (18)	13 (21)	7 (8)	125 (172)
GA	191 (282)	1 (6)	18 (24)	2 (24)	212 (315)
HU	74 (166)	3 (11)	5 (15)		82 (192)
IT	157 (186)	109 (133)			266 (319)
NL	256 (458)	29 (89)	28 (78)	3 (13)	326 (638)
PT	27 (51)		9 (11)		36 (62)

Tab. 1 Number of definitions per POS tag per language linked to a BabelNet synset.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

In Table 1 we can see how many definitions per POS tag we were able to link to a BabelNet synset in our gold standard. In parenthesis we indicate the definitions which had no match.

3 Results

When evaluated on the inference task, identifying negative examples as well, we obtain the following results.

		BG	ES	ET	GA	HU	IT	NL	PT	AVG
Cross Encoder	**F1**	76.39	**55.50**	76.92	**45.13**	**66.67**	78.75	69.53	31.11	62.50
	Rec	86.84	69.32	73.91	**65.32**	73.97	**84.75**	85.34	22.58	70.25
	Prec	68.18	**46.28**	80.19	34.47	**60.67**	73.54	58.66	50.00	**59.00**
	Acc	70.26	68.90	70.35	37.46	**71.88**	68.03	68.81	50.00	63.21
ESCHER	**F1**	**80.09**	54.42	**78.26**	44.97	57.69	**79.66**	**69.88**	54.17	**64.89**
	Rec	**93.37**	**72.64**	**79.12**	58.46	**78.95**	84.30	**86.34**	50.00	**75.40**
	Prec	**70.12**	43.50	77.42	**36.54**	45.45	75.50	**58.68**	**59.09**	58.29
	Acc	**74.39**	**70.00**	**73.15**	**38.21**	71.43	**69.43**	**71.83**	**61.40**	**66.23**
Embedding based	**F1**	73.78	49.29	76.56	17.32	64.20	75.50	66.67	**57.14**	60.06
	Rec	85.95	58.16	68.97	11.76	73.24	72.15	72.18	**57.14**	62.44
	Prec	64.63	42.77	**86.02**	32.84	57.14	**79.17**	61.94	57.14	60.21
	Acc	67.06	68.12	71.51	33.33	69.79	65.20	69.91	61.29	63.28

Tab 2. Performance evaluation on positive and negative examples.

We can see how the ESCHER achieves a better overall performance, except in terms of precision. However F1 is two points higher than the Cross-encoder and almost 5 points over the embedding-based model.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

When evaluating only on senses for which there is a match in BabelNet, we consider only the accuracy score.

	BG	ES	ET	GA	HU	IT	NL	PT	AVG
Cross Encoder	81.98	67.08	85.60	**44.81**	**82.93**	78.20	78.80	**61.11**	**72.56**
ESCHER	**85.51**	**69.55**	79.21	44.33	78.26	**80.00**	**79.57**	54.55	71.37
Embedding Based	78.83	61.80	**86.40**	36.79	80.49	76.69	75.32	52.78	68.64

Tab 3. Performance evaluation on senses linked to BabelNet.

In this case the cross-encoder achieves a better performance, one point over ESCHER. We can see how in both evaluations there are differences across languages, perhaps due to the pre-trained models that each model is based on. ESCHER (based on XLM-roberta) has a much better performance for Bulgarian in both setups, while Hungarian was better on the cross-encoder (based on mDeberta).

4 TIAD Shared Task

Inducing new translation pairs across dictionaries is an important task that facilitates processing and maintaining lexicographical data. NUIG team participate and describe their submissions to the Translation Inference Across Dictionaries (TIAD) shared task of 2021. In the shared task, the datasets provided this year contain 44 languages and 53 language pairs, with a total number of 1,540,996 translations between 1,750,917 lexical entries[1].

NUIG systems mainly rely on the MUSE and VecMap cross-lingual word embedding mapping to create new translation pairs between English, French and Portuguese data. The team also created two regression models based on the graph analysis features. The details are described in 4.1.

4.1 System's methodology

[1] https://tiad2021.unizar.es/task.html

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

(i) Graph-based regression models

Our graph-based methods are based on the analysis that was performed previously in McCrae and Arcan, where the algorithm for extracting the connections between two nodes was applied as previously. We further extended this algorithm to extract the following measures from the graph:

- dmin(n, m): The minimum distance in the graph between the two nodes.
- N∗(n, m): The number of paths between the nodes of any length.
- N2(n, m): The number of paths between the nodes of length 2.
- a∗(n): The number of nodes reachable from node n. – a1(n): The number of nodes directly connected to node n

We used dmin, N∗ and N2 directly as features in our system and we added to methods based on the One-Time Inverse Consultation[add citation] as follows:

$$N*(m, n)/a*(n)a*(m)$$

$$N2(m, n)/a1(n)a1(m)$$

This leads to five features in total which could be combined as a linear model. Given that no training data is provided in the task, we apply our graph based approach on the English-Spanish translation pairs to extract features for training. This data set is then used to train two Support Vector Regression1 models with a linear kernel, namely ULD graphSVR and ULD OnetaSVR. Given a new data instance based on our target languages, the regression models predict a score corresponding to the confidence score of the shared task. It is worth noting that all features and normalized and scaled properly

(ii) Cross-lingual embedding mappings

One major limitation of graph-based methods is due to the limited coverage of connectivity between certain translations, i.e. nodes. It illustrates some of the translations that can be retrieved for the word ‘chaotic’ (adjective) in the Apertium translation graph where the Portuguese translation ‘ca´otico’ (‘chaotic’) is not retrievable by traversing intermediate nodes.

In order to tackle this limitation, we use two unsupervised cross-lingual word embedding mapping techniques, namely VecMap and MUSE. These techniques find a mapping between the monolingual

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

word embedding spaces of the source and target languages. It shows a visualization of 'chaotique' ('chaotic') in French and its closest words in both the French and Portuguese vector spaces.

VecMap based cross-lingual embedding was built on the unsupervised method using pre-trained French and English fastText monolingual embedding models2. After building the cross-lingual embedding and achieving confidence scores, we used monolingual pre-trained UDPipe 2.5 models to generate the part-of-speech features only of the target(French) language. Furthermore, the generated parts-of-speech tags were mapped with parts-of-speech tags of the shared task.

In the same vein, a mapping is learned using the MUSE unsupervised method and fastText monolingual embeddings of French, English and Portuguese which takes use of adversarial learning followed by iterative Procrustes refinement (default configuration of n refinements = 5)4. Ultimately, these mappings are to create new translation pairs between the 10 most nearest translations in the target language using cosine similarity. The cosine similarity score is then considered as the confidence score in the final submission and the part-of-speech of the source word is used for the target predictions as well.

4.2 Evaluation

System	Precision	Recall	F1-measure	Coverage
ULD GraphSVR	0.7	0.49	0.57	0.69
baseline-Word2Vec	0.69	0.23	0.33	0.4
ULD MUSE	0.29	0.41	0.33	0.65
baseline-OTIC	0.78	0.18	0.29	0.28
ULD OnetaSVR	0.76	0.1	0.17	0.14
ULD Oneta	0.64	0.07	0.13	0.11
ULD VecMap	0.36	0.01	0.01	0.02

Tab. 4 The performance of NUIG-ULD systems in comparison to the baselines.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

The submitted systems perform above the baseline systems. Results are averaged for every system and correspond to an arbitrary 0.5 threshold. In addition to the ULD Graph_SVR system, the ULD_MUSE system covers over half of the dictionary entries.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

4 API Documentation

In this section, we provide detailed documentation regarding the usage of our web service. The present API is made up of the following three components:

a) **Model**: This module contains the model used to perform cross-lingual dictionary linking.
b) **BabelNet**: This module contains the code used to obtain the BabelNet definitions of a given lemma and POS tag.
c) **REST API**: This module contains the code that will be used to obtain the dictionary senses from LEXONOMY of a given lemma and pos tag and communicate with the other two modules to perform the linking task.

The BabelNet module can be run independently, while the REST API depends on the BabelNet one.

4.1 a) Model

The model is loaded when inference is performed on the pending requests. This is dealt by the backend with a cronjob. More details in the API section. For the docker container to access the model files, please place them in the model/ directory at the root of the project.

b) BabelNet module

This module allows us to retrieve definitions from BabelNet.

c) REST API

The REST API uses python FastAPI, as well as a Pydantic model to validate the input, manage a database (sqlite) of the requests and to run them asynchronously. In fact, when a request is submitted, an ID is returned to the user, which can be used to check the status of the request. Once the request has been completed, the user can obtain the results using the ID. This component is dockerized. To build the container please run:

GPU:

docker build -f dockerfiles/Dockerfile --build-arg MODEL_PATH="model" -t dict_api .

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

And then to run it:

PO*RT=12345*

docker run -p $PORT:80 --name dict_api --gpus all dict_api

The port variable can be set to whatever is needed.

CPU:

docker build -f dockerfiles/Dockerfile.cpu --build-arg MODEL_PATH="model" -t dict_api .

And then to run it:

PORT=12345

docker run -p $PORT:80 --name dict_api dict_api

The port variable can be set to whatever is needed.

Cronjob:

To run the inference script we will need to run a cron job that will trigger the inference on the requests made that day:

docker exec dict_api bash -c 'python3 run_async.py'

4.2 References

[1] ***Agirre, E., De Lacalle, O. L., Fellbaum, C., Hsieh, S. K., Tesconi, M., Monachini, M., ... & Segers, R.*** (2010, July). Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In Proceedings of the 5th international workshop on semantic evaluation (pp. 75-80).

[2] ***Barba, E., Procopio, L., Campolungo, N., Pasini, T., & Navigli, R.*** (2021, January). Mulan: Multilingual label propagation for word sense disambiguation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence* (pp. 3837-3844).

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

[3] Barba, E., Pasini, T., Navigli, R. (2021, January) ESC: Redesigning WSD with Extractive Sense Comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

[4] ***Devlin, J., Chang, M. W., Lee, K., & Toutanova, K.*** (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

[5] ***Navigli, R., & Ponzetto, S. P.*** (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial intelligence, 193, 217-250.

[6] https://github.com/elexis-eu/MWSA

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.