D3.1 Research Challenge Report v1

Author(s): Christian Chiarcos, Philipp Cimiano, Christian Fäth, Thierry Declerck, Basil Ell, Jorge Gracia, John McCrae, Maria Pia di Buono, Mariano Rico, Housam Ziad, Paul Buitelaar

Date: 27.09.2019

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825182. 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.

H2020-ICT-29b

Grant Agreement No. 825182

Prêt-à-LLOD - Ready-to-use Multilingual Linked Language Data for Knowledge Services across Sectors

D3.1 Research Challenge Report

Deliverable Number: D3.1
Dissemination Level: Public
Delivery Date: 2019-09-30
Version: 1.0
Author(s): Christian Chiarcos, Philipp Cimiano, Christian Fäth, Thierry Declerck, Basil Ell, Jorge Gracia, John McCrae, Maria Pia di Buono, Mariano Rico, Housam Ziad, Paul Buitelaar

Document History

Version Date	Changes	Authors
2019-07-19	**Initial document**	**Christian Chiarcos / Christian Fäth**
2019-09-21	**Merged text from task reports**	**Christian Fäth (text from all authors)**
2019-09-23	**Consolidated overall layout, references; minor textual adjustments; moved parts to appendix; added intro.**	**Christian Fäth**
2019-09-26	**Revisioning of text**	**All authors**
2019-09-27	**Revisioning of full document**	**Christian Fäth**
2019-09-30	**Final revisioning**	**John McCrae**

Table of Contents

D3.1 Research Challenge Report v1

1. Introduction

The Prêt-à-LLOD project aims at creating a data value chain (Figure 1) for linguistic linked open data (LLOD)[1] to be used across industrial sectors within the emerging Digital Single Market in Europe. Working with linguistic data can be a very time-consuming process. Discovering the resources across existing repositories and endpoints is not the only challenge a user is faced with. After overcoming the hurdles of licensing and gaining access to the required datasets, heterogeneous data models, formats and annotation schemes with varying levels of detail may require complex transformation and linking processes in order to extract the pieces of information relevant to a specific research topic. Furthermore, existing Natural Language Processing tools require very specific input data making intermediate transformation steps necessary within complex workflows. Prêt-à-LLOD tackles these challenges by developing software components for each of them and placing them in the context of four industrial pilot projects to evaluate their usability.

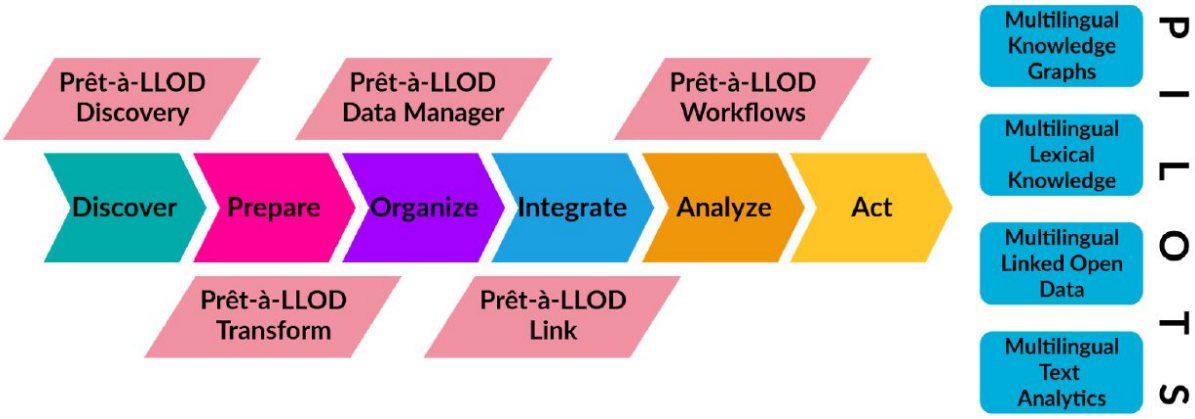


Figure 1: Prêt-à-LLOD data value chain

While the development of vocabularies and community standards as well as the discovery and license management of linguistic resources is pursued in other work packages (mainly WP5), within WP3 three research challenges are tackled by three respective tasks, each responsible for one software component:

- In **T3.1**, **Prêt-à-LLOD Transform** addresses the challenge of "Transforming language resources and language data. Methodologies will be developed for the transformation of language resources and language data into LLOD representations."
- In **T3.2**, **Prêt-à-LLOD Link** addresses the challenge of "Linking conceptual and lexical data for language services. Novel (semi-)automatic methods will be studied that aim at establishing links across multilingual LLOD datasets and models."

[1] For details about the Linguistic Linked Open Data cloud as a subgraph of the LOD cloud, see http://linguistic-lod.org/

- In **T3.3**, **Prêt-à-LLOD Workflows** addresses the challenge to create “Workflows for Portable and Scalable Semantic Language Services. A protocol, based on semantic markup, will be developed to enable language services to be easily connected into multi-server workflows.”

In this report we describe the current state of development of these three software components. In section 2 we describe data transformation using FINTAN, the flexible integrated transformation and annotation engineering platform and outline its specifications and usability with three specific case studies. In section 3 we introduce services for interlinking multilingual resources on the levels of ontology lexicalization (A), ontology matching (B) and lexical data (C) using the Naisc, GAAS and Lemonade tools. In section 4, we introduce Teanga as our primary workflow management tool for NLP services.

2. Transformation [T3.1]

2.1 Motivation

In order to prepare resources for use within the Prêt-à-LLOD project, and especially in order to fulfill the project goals of supporting 50 input formats and making available 1000 resources as LLOD, Task 3.1 aims at creating a generic framework for transforming resources to RDF. The main challenges herein stem from the vast amount of heterogeneous resources to be dealt with in the project and the industrial pilots. Since a second goal is the normalization of language resources to predefined target formats and existing community standards (T5.1), the transformation goals are not only subject to quantitative assessment but also must be able to meet qualitative requirements.

One feasible approach would be the creation of a monolithic but mostly generic converter which is able to produce baseline RDF for use in further tasks. This would bear the advantage of easily being able to meet quantitative requirements but also bear the risk of lacking data quality. Apart from that, converters like this already exist, e.g. CSV2RDF (Tandy et al., 2015), R2RML (Das et al., 2012) but their output is not very easily adoptable for linguistic use cases. Furthermore, to some extent these formats tend to reflect the original data storage paradigm of their source material within RDF and therefore generate unnecessary overhead while not taking sufficient advantage of the native graph layout.

Instead, since data types relevant for Prêt-à-LLOD mainly comprise dictionaries and corpora our primary target models will be OntoLex-Lemon (McCrae et al., 2012; Cimiano et al., 2016) as well as NIF (Hellmann et al., 2013), CoNLL-RDF (Chiarcos and Fäth, 2017) and POWLA[2] (Chiarcos, 2012) respectively. These formats are well established and widely used within the LLOD community. This will aid in creating resources which are both linguistically rich and reusable across work packages and beyond the scope of the project.

However, the transformation steps needed to fully convert existing heterogeneous resources into these target models will be far more complex than the simple RDF rendering approaches described above. By creating several monolithic but highly resource-specific converters, we could easily meet qualitative requirements but might never be able to catch up on the quantitative goals.

We therefore decided upon a more flexible approach: by creating a modular framework of interoperable transformation steps, we will be able to combine the best of both worlds by creating simple baseline RDF converters (or integrating existing ones) and enriching their output using graph transformation to meet the qualitative requirements of desired target models, thus increasing reusability. Certain source material might only need some intermediate steps or minor adjustments to existing modules to be transformed into valid

[2] POWLA has recently become one of the most relevant formats for representing corpora with more complex semantic or syntactic structures. (Cimiano et al., 2019)

RDF resources. On the other hand, only some additional graph transformation steps might be necessary to support additional target models.

In the following sections we will first describe a set of case studies which we are currently performing building upon the project partners' software stack. From these case studies we then derive the theoretical basis and requirements for the Flexible and Integrative Transformation and Annotation eNgineering (FINTAN) platform which will be the main contribution to the T3.1 software deliverable.

2.2 Case study: Open Multilingual Wordnet in TSV

Wordnets are well-established lexical resources with a wide range of applications in various Natural Language Processing (NLP) fields, like Machine Translation, Information Retrieval, Query Expansion, Document Classification, etc. (Morato et al., 2004). For more than twenty years they have been elaborately set up and maintained by hand, especially the original Princeton WordNet of English (PWN) (Fellbaum, 1998). In recent years, there has been an increasing amount of activities in which open wordnets for different languages have been automatically extracted from other resources and enriched with lexical semantics information, building the so-called Open Multilingual Wordnet (OMW) (Bond and Paik, 2012), which is merging more than 35 open wordnets that are linked through the Collaborative Interlingual Index (CILI) (Bond and Foster, 2013; Bond et al., 2016).

As stated on the web page of OMW, the listed wordnets are of different quality, and some of them were in fact extracted from different types of language resources. OMW provided for some corrections and for a harmonization of the resources, and published them in a uniform tabular format (Tab Separated Values, TSV), which is displayed below, exemplified here by entries from the Italian OMW resource:

```
08388207-n ita:lemma nobiltà
08388207-n ita:lemma aristocrazia
08388207-n ita:lemma patriziato
08388207-n ita:def_0 l'insieme degli aristocratici
08388207-n ita:def_1 l'insieme dei nobili
...
14842992-n ita:lemma terra
14842992-n ita:lemma terreno
14842992-n ita:lemma suolo
14842992-n ita:def_0 parte superficiale della crosta terrestre sulla quale si
      sta o si cammina 14842992-n ita:exe_0 si piegò con fatica per
      raccogliere da terra i sacchetti, pronta a salire sull'autobus
14842992-n ita:exe_1 il tizio comincio' a rotolarsi per terra in preda a
      dolori lancinanti
```

In the two examples displayed just above, the uniform tabular format of OMW is delivering information on the synset IDs (08388207-n and 14842992-n), which are including the part-of-speech ("n") of the associated lemma(s). The nominal lemmas associated with the synset-ID 08388207-n are "nobiltà" (*nobility*), "aristocrazia" (*nobility, aristocracy*) and

"patriziato" (*aristocracy*). The nominal lemmas associated with the synset-ID 14842992-n are "terra" (*earth, land, soil*), "terreno" (*ground, terrain, soil*) and "suolo" (l*and, earth, ground*). If available, definitions ("glosses") are provided (marked with the feature **`ita:def`**), as well as examples (marked with the feature **`ita:exe`**). This tabular format is used for all the OMW data sets. This makes it relatively straightforward to map OMW data to OntoLex-Lemon (cf. Figure 2).

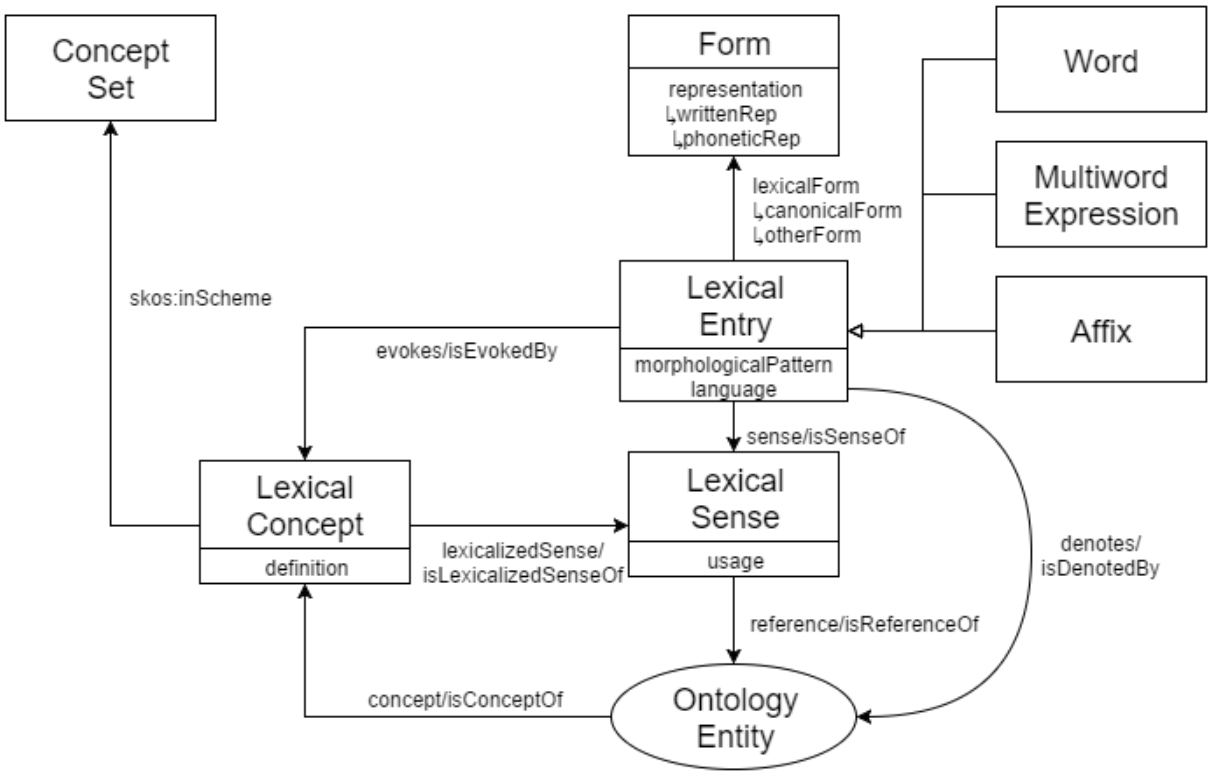


Figure 2: The core module of OntoLex-Lemon: Ontology Lexicon Interface. Graphic taken from https://www.w3.org/2016/05/ontolex/.

In a first step, we concentrated on three languages included in OWW: French, Italian and Spanish. We are using "WOLF (Wordnet Libre du Français)" for French, "ItalWordNet" for Italian and "Multilingual Central Repository" for Spanish (this resource contains also wordnets for Catalan, Basque and Galician languages).[3] A Python script was implemented for porting the OMW data sets to OntoLex-Lemon.[4] A design decision was to extract only the synset information and to encode the synsets as instances of the LexicalConcept class of OntoLex-Lemon. As we expect to have the OMW lemmas present in already existing lexicons, we will in a next step just link the synsets to those lemmas, which are encoded as instances of the OntoLex-Lemon LexicalEntry class. This way we achieve a higher level of modularity. Since the synsets are now encoded as instances of the LexicalConcept class,

[3] See respectively (Sagot and Fišer, 2008), (Pianta et al., 2002; Toral et al., 2010) and (Gonzalez-Agirre et al., 2012).

[4] The simple Python script was implemented for our experimental purposes. In the future, we will make use of and adapt the web service made available by NUIG for transforming WordNets in various formats, like JSON-LD or a serialization of RDF (see https://globalwordnet.github.io/schemas/)

each synset-ID gets a Unique Resource Identifier (URI), and does not have to be repeated for each lemma it is associated with, but can just link to those via the OntoLex-Lemon property **`isEvokedBy`**, as this can be seen in Figure 2. This way we have also a more compact (graph-based) representation as in the original representation of the OMW data.

In the listing below we show examples of the OntoLex-Lemon encoding of two synsets for Spanish with their corresponding lemmas:

```
:synset_spawn-13491616-n
      rdf:type ontolex:LexicalConcept ;
      ontolex:isEvokedBy :lex_cura-13491616-n ;
      skos:inScheme :spawnet .


:synset_spawn-10470779-n
      rdf:type ontolex:LexicalConcept ;
      ontolex:isEvokedBy :lex_cura-10470779-n ;
      skos:inScheme :spawnet .

:lex_cura-13491616-n a ontolex:LexicalEntry ;
      lexinfo:gender lexinfo:masc;
      lexinfo:partOfSpeech lexinfo:noun ;
      ontolex:evokes :synset_spawn-13491616-n ;
      ontolex:canonicalForm :form_cura ;
      ontolex:otherForm :form_cura_plural .

:lex_cura-10470779-n a ontolex:LexicalEntry ;
      lexinfo:gender lexinfo:fem ;
      lexinfo:partOfSpeech lexinfo:noun ;
      ontolex:evokes :synset_spawn-10470779-n ;
      ontolex:canonicalForm :form_cura ;
      ontolex:otherForm :form_cura_plural .
```

The lemmas associated with these synsets are “cura”. In the original OMW, the lemmas are just literals and not real lexical entries, associated with more complex linguistic information, beside part of speech. We add in Listing 1 the two lexicon entries we have defined for “cura” (one for the meaning “priest”, used in masculine gender, and one for the meaning ”cure”, used in feminine gender).

In addition to this transformation, we aim at enriching the OMW data sets already encoded in OntoLex-Lemon with further morphological semantic information. For this we already mapped the French, Italian and Spanish morphological resources included in the Mmorph data sets (Petitpierre and Russell, 1995) into OntoLex-Lemon (Declerck and Racioppa, 2019) and in doing so, we are bridging/linking the two types of data sources.[5]

[5] Work will also be dedicated for mapping the OMW data to the OntoLex-Lemon representation that is made available by the transformation of the Universal Morphologies (see Section 2.3)

2.3 Case study: Universal Morphologies

The Universal Morphology (UniMorph) project provides a universal way to annotate morphological data in a universal schema. This allows an inflected word from any language to be defined by its lexical meaning, typically carried by the lemma, and by a rendering of its inflectional form in terms of a bundle of morphological features from the UniMorph annotation schema (Sylak-Glassman et al. 2015).

In context of LLODifier[6], a larger toolset for transforming linguistic data into a shallow Linked Data representation, we already provided a transformation suite for mapping UniMorph data to OntoLex-Lemon (Chiarcos et al. 2018a), using our well-established CoNLL-RDF library (Chiarcos et al. 2018b, Chiarcos and Fäth 2017). Though CoNLL-RDF was originally built for transforming corpora into an isomorphic RDF representation, it was applicable to the dictionary-type UniMorph data out-of-the-box mainly because of their simple layout and TSV structure. This makes it an ideal case study for testing CoNLL-RDFs streamed graph transformation capabilities on different types of data.

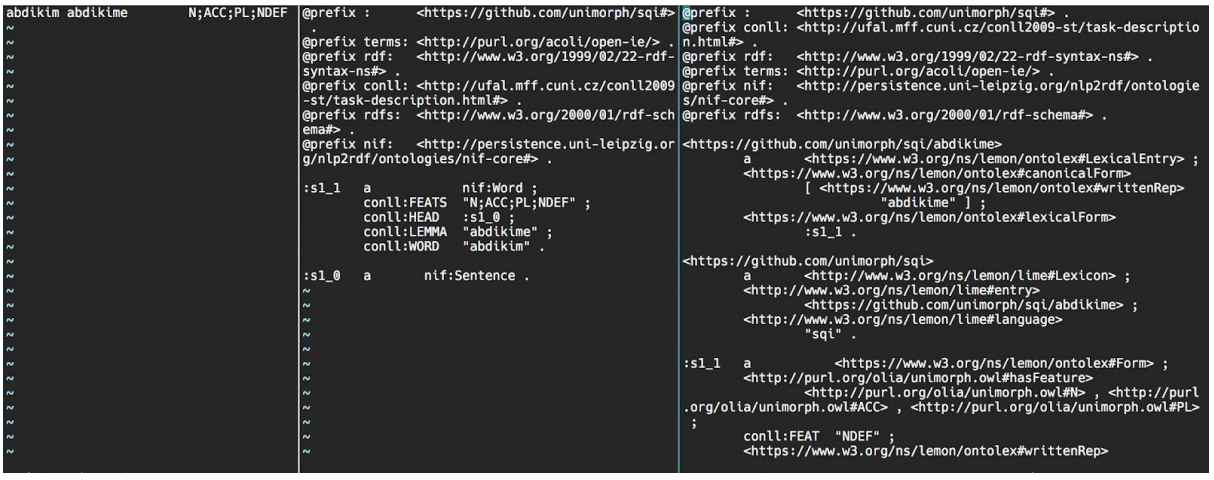


Figure 3: Transformation process example for a single entry.

The first challenge to overcome was the transformation of a CoNLL-RDF corpus representation into an actual OntoLex-Lemon dictionary. Figure 3 visualizes the three-step process of the transformation. The left column represents a single entry in the original UniMorph dictionary for Armenian. Using CoNLL-RDF, we transform it to a shallow representation in RDF seen in the middle column. As CoNLL-RDF was initially tailored towards corpus data, we see concepts as words and sentences, these are however irrelevant for our approach and are removed subsequently. Using the `CoNLLRDFUpdater` we apply the actual graph transformation of the shallow corpus-like CoNLL-RDF representation into a valid OntoLex-Lemon representation. The right column shows the transformed data. In the process, the morphological features have been linked to OLiA (Chiarcos and Sukhareva, 2015) concepts, except for the NDEF feature. Because the linked resource did not contain this entry, we fall back on the representation as an entry in the CoNLL feature column.

[6] The LLODifier tools are available at https://github.com/acoli-repo/LLODifier/

The second challenge is a significant scalability issue. CoNLL-RDF was designed to efficiently stream even extremely large corpora sentence by sentence. This would limit both the amount of memory consumed as well as the processing complexity of SPARQL updates since they would be applied only to single sentences instead of a giant monolithical graph. Our recent improvements made to the CoNLL-RDF library enable parallelization, which can significantly speed up this process.

Since UniMorph does not have sentence borders, the whole corpus was originally treated as one `nif:Sentence` with each `LexicalEntry` rendered as a `nif:Word` with respective annotations. This would limit the processable size of the dictionary to the amount of available RAM. Yet, it established an ideal testing ground for the performance scalability of the streamed and parallelized SPARQL updates used in CoNLL-RDF and planned for FINTAN. In this case study we redo our previous experiment with the improved package to demonstrate the effectiveness of these improvements.

We tested three approaches to convert the UniMorph data to LLOD. The first approach reads the entire UniMorph file *en bloc*, applies the transformations and writes the output. This leads to non-redundant RDF with the smallest possible file size, as dictionary metadata is not repeated for each entry. However, memory limits do apply for very large dictionaries. In this case, splitting the entries and transforming each entry separately becomes necessary.

This is done in the second approach: Using the `CoNLLStreamExtractor` module of the CoNLL-RDF package we process each UniMorph entry individually. This increases the size of the resulting file significantly, because prefixes and dictionary metadata are repeated for each entry. Furthermore, the processing takes significantly more time, as the OLiA model to be linked to needs to be loaded for each sentence. This was the state of CoNLL-RDF during the original publication of the UniMorph converter.

In the final approach, using our improved `CoNLLRDFUpdater` module, we cache the OLiA model so it only needs to be loaded once at the startup time of the pipeline. Furthermore, we use multithreaded processing so single entries can be processed independently in parallel. This allows us to transform data of any size with highly increased processing speed.

Table 1: Time and size comparison of the three processing approaches.
(Performed on a 3.79 GHz i5 quadcore with 16 GB of memory.)

	En bloc whole file	**Line-wise single thread**	**Line-wise multithread**
Transformation time	6m24s	12m34s	3m00s
# of OLiA loads	1	33484	1
# of lines in output file	449308	940922	940922

Table 1 shows the results of our experiment. While the *en bloc* approach is fairly fast, it is susceptible to size limitations. The line-wise processing without parallelization and precaching of external resources is much slower due to the constant loading and unloading of the OLiA models.

The line-wise multithread approach not only removes the loading penalty by precaching, it also displays that our stream-based graph transformation outperforms established database engines (here Apache JENA) in specific use cases. In the *en bloc* approach, transformation is achieved by a single SPARQL update executed on the whole dictionary while the database engine is distributing memory and processing power. In the multithread approach we distribute processing power across all cores executing the same update multiple times but only on a single `LexicalEntry` resulting in much higher performance.

However, the issue of verbose and chunked result data remains. This is a limitation of the current, corpus-oriented CoNLL-RDF implementation and could be removed by a very simple postprocessing step. Since FINTAN will be developed with a wider range of language resources in mind, these limitations will no longer apply to the new implementation. The performance and scalability achieved with CoNLL-RDF on a non-corpus dataset in this case study show the capabilities of the prospective FINTAN architecture.

2.4 Case study: Transforming terminological data

Terminologies are also an important resource for the use cases described in Prêt-à-LLOD, as they offer domain specific "lexicalization" of concepts, realised as (sometimes multilingual) terms. For the transformation of terminologies in OntoLex-Lemon in a first step we consider those terminologies that are available in a standard representation, in our case TBX (Term Base eXchange),[7] a format that can be mapped onto OntoLex-Lemon without losing information. A specification for converting TBX to RDF[8] was already available in 2015, before the final specification of OntoLex-Lemon (Cimiano et al., 2016) was published. We can now port TBX into the final version of OntoLex-Lemon, and thus make terminological data available in the Linguistic Linked Open Data cloud. A full description of the mapping is displayed in Appendix A.

A first set of TBX resources we are considering for the transformation into OntoLex-Lemon is the data listed in ELRC-SHARE (https://elrc-share.eu/). In the following we quote from Lösch et al. (2018),[9] for introducing to the ELRC-SHARE portal:

"In order to help improve the quality, coverage and performance of automated translation solutions for current and future Connecting Europe Facility (CEF) digital services, the European Language Resource Coordination (ELRC) consortium was set up through a service contract operating under the European Commission's CEF SMART 2014/1074

[7] See https://www.tbxinfo.net/ for more information.
[8] https://www.w3.org/2015/09/bpmlod-reports/multilingual-terminologies/.
See also (Cimiano et al., 2015)
[9] See http://lr-coordination.eu/sites/default/files/Ireland2/2.3_Preparing%20and%20sharing%20data%20with%20the%20ELRC%20repository.pdf for more details.

programme to initiate a number of actions to support the collection of Language Resources (LRs) within the public sector in EU member and CEF-affiliated countries. [...] In order to gather resources shared by the contributors, the ELRC-SHARE Repository was set up. [...] The collected LRs cover all official EU languages, plus Icelandic and Norwegian."

While the collected language resources in ERLC-SHARE are meant to support primarily the eTranslation platform,[10] a relevant number of the collected data is directly relevant for Prêt-à-LLOD. We think primarily of all the ELRC-SHARE resources classified as "lexical conceptual resources" (including terminology), which can be mono- or multilingual. Most of the lexical conceptual resources in ELRC-SHARE are encoded in TBX. These resources are mostly relevant to one (or more) European Digital Service Infrastructure (DSI).[11] Use cases of Prêt-à-LLOD are closely related to some of the DSIs, like eHealth, Business Registers Interconnection System (BRIS), Public Open Data, or Electronic Exchange of Social Security Information (EESSI).

In the future we will implement a reverse transformation: terminological datasets that are available in the LLOD cloud can be transformed into TBX and so added to the ELRC-SHARE repository.

2.5 Flexible integrated transformation and annotation engineering (FINTAN) platform

Keeping in mind the aforementioned case studies and the wealth of heterogeneous input formats we designed the FINTAN platform as a pool of interoperable transformation modules which can be interconnected to render virtually any input to any output format with minimum effort. As mentioned in the motivation section, our goal cannot be to simply generate a shallow RDF representation which closely reflects the data structure of the respective source data but we need to aim at using existing standards and data models like OntoLex-Lemon, NIF, CoNLL-RDF or POWLA as our primary target formats. This however will require much more complex transformation steps.
In order to render these steps in a unified fashion and make them reusable across transformation modules, FINTAN will make heavy use of SPARQL updates to transform the RDF output of shallow converter modules into valid datasets adhering to the target models. Since SPARQL is typically executed on triple stores containing full datasets, respective processing modules would require vast amounts of memory to process larger datasets. With CoNLL-RDF, we already have a working infrastructure for targeting this limitation. CoNLL-RDF is designed to process very large CoNLL corpora by applying SPARQL updates sentence by sentence in a parallelized fashion. This stream-based approach not only eliminates the memory limitation but has the potential of also increasing the processing speed by better distribution of transformation steps across threads. By design, CoNLL-RDF is currently limited to corpora in TSV format. The Unimorph case study mentioned above showcases the potential speed improvements and represents a first attempt to apply the

[10] See https://ec.europa.eu/info/resources-partners/machine-translation-public-administrations-etranslation_en for more details.
[11] See https://ec.europa.eu/inea/en/connecting-europe-facility/cef-telecom/projects-by-dsi for a listing of such DSIs.

existing corpus infrastructure to simple dictionaries. Yet, it is still limited to TSV format and induces some corpus specific overhead into the resulting OntoLex-Lemon dictionary.

With FINTAN we tackle these limitations and will create a universally applicable resource transformation tool. While non-TSV source formats can be addressed by the integration of existing RDF converters as processing modules, the sentence-based streaming will be replaced by a more generic block-based approach which will allow e.g. dictionaries to be transformed per **`LexicalEntry`**.

Since SPARQL updates may become very complex, might use external reference resources (e.g. DBPedia for wikification) and could also be applied iteratively across multiple processing blocks, we additionally intend to create a SPARQL update development frontend to aid the implementation process. SPARQL update development will be assisted by a SPARQL editor consisting of an input pane for textual editing and a visualization engine which draws a graph derived from the update displaying **`INSERT`**, **`DELETE`** and referenced nodes. This tool will be especially helpful for maintaining more complex update scripts.

Apart from that, a metadata format will be developed encapsulating mandatory prerequisites for resources this update can be applied to. This may include: data model (e.g. OntoLex-Lemon), property constraints, execution requirements (recursive, single, needs precached resource). This metadata will be derived from the script as far as possible. Furthermore, some updates, which are closely tied together, can be grouped so they appear as one module. Since T3.3 is facing a similar challenge to render interdependencies of Teanga modules, our metadata formats may share major parts of their vocabularies to further increase interoperability.

Furthermore, since modules may require specific libraries and may be written in any programming language (e.g. Java for CoNLL-RDF, Python for the OMW case study) it is necessary to build a workflow management engine which is capable of interconnecting and running any kind of modules to transform a stream of data. With Task 3.3 addressing similar challenges with the Teanga platform we will evaluate options for a unified graphical interface and possibly also build on the underlying docker architecture for FINTAN. This will not only increase efficiency within WP3 but will also result in a more streamlined user experience.

For validating the resulting pipelines we will include validation measures for the most common target formats and models (in our case OntoLex-Lemon, CoNLL-RDF, NIF and POWLA). This will also include RDF validation for at least TURTLE serializations. Should we need to provide output support for other formats (like native CoNLL etc.) we will integrate external validators as modules, if possible. If no approved validator exists, we might not be able to cover full validation within the project scope.

After a pipeline has been configured, we will provide several runtime and deployment modes. Debugging will allow a user to run a pipeline locally and assess preliminary results of intermediate steps for finding mistakes. Fully functional converter pipelines can then be deployed as integrated containers with configuration files for integration into Teanga.

Figure 4 displays the main components of FINTAN:

- In addition to an I/O frontend which will allow the user to upload or manage his data, the development environment will build on a common workflow management UI with Teanga and also integrate a frontend for SPARQL development.
- The processing modules will be made available as a pool for the workflow manager. They may comprise shallow converters, validators, serializers etc.
- The SPARQL Updater will be based on the existing CoNLL-RDF infrastructure and will work as a large module with singular SPARQL scripts as submodules. It will be used to process complex and resource-heavy transformation steps in a parallelized fashion.
- Export functionalities will not only include the data in their respective output formats, but also the pipelines themselves can be exported as configuration files or even wrapped as singular containers for Teanga.

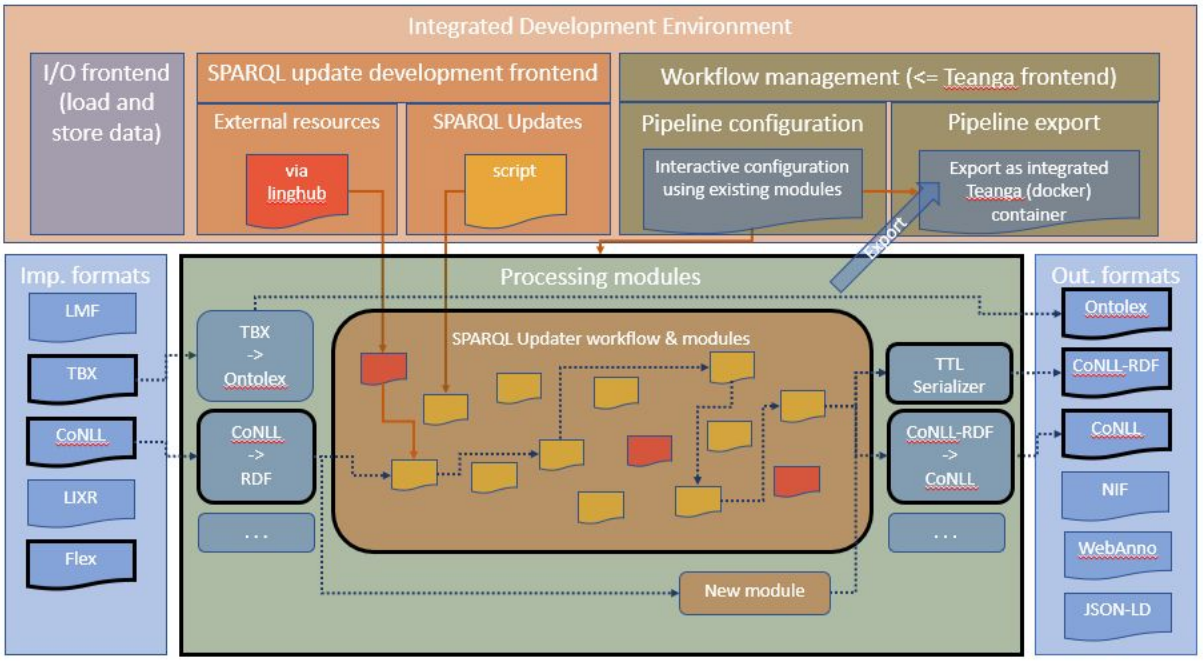


Figure 4: Prospective FINTAN architecture

Within the project, the FINTAN platform will not provide data through a black box maintained by GUF[12] but will also allow other project partners, especially from WP4 to contribute their existing converters as modules or to make adjustments to pipelines on their own. In its final state, the system will be able to transform many kinds of resources into common data models applying task specific annotations. Additionally, converter pipelines can be deployed to T3.3's Teanga platform to make them publicly available, thus enabling RDF-based NLP modules to directly feed on generic resource types, further increasing scope and applicability for long term use by a wider audience.

[12] The Prêt-à-LLOD project partner Johann Wolfgang Goethe-Universität Frankfurt a. M.

3. Linking [T3.2]

3.1 Motivation

Challenge 4 of Prêt-à-LLOD ("Linking conceptual and lexical data for language services") is addressed by the "Prêt-à-LLOD Linking" technical component. The development of such a component is carried out in the context of task T3.2, whose progress is described in this part of the document. In this task, novel (semi-)automatic methods are studied, aiming to establish monolingual and cross-lingual links across LLOD datasets and models.

To that end, state-of-the art similarity and relatedness measures will be adapted to linked data, in particular to exploit the variety and richness of linguistic features found in the LLOD cloud. This task considers emerging techniques based on word embeddings and deep learning, jointly with knowledge-based and distributional semantics-based ones. In this task, research is being done on LLOD-based models, methods, and techniques for accessing and exploiting data across different languages. Further, methods for lexicalizing existing ontologies in multiple languages are also studied in this task, by linking them to lexical resources and implementing APIs that provide access to this knowledge. This task will provide the linking technology that underpins many of the pilots and the discovery mechanisms in WP5.

In this task, links at three different levels are being analysed:

Level A: links between the conceptual and lexical level (lexicalisation). In that case, links are established between concepts and their lexical realisations, e.g, through the **`ontolex:reference`** property. See the example in Figure 5, where two ontologies in EN and FR respectively are lexicalised through their corresponding OntoLex-lemon lexicons.

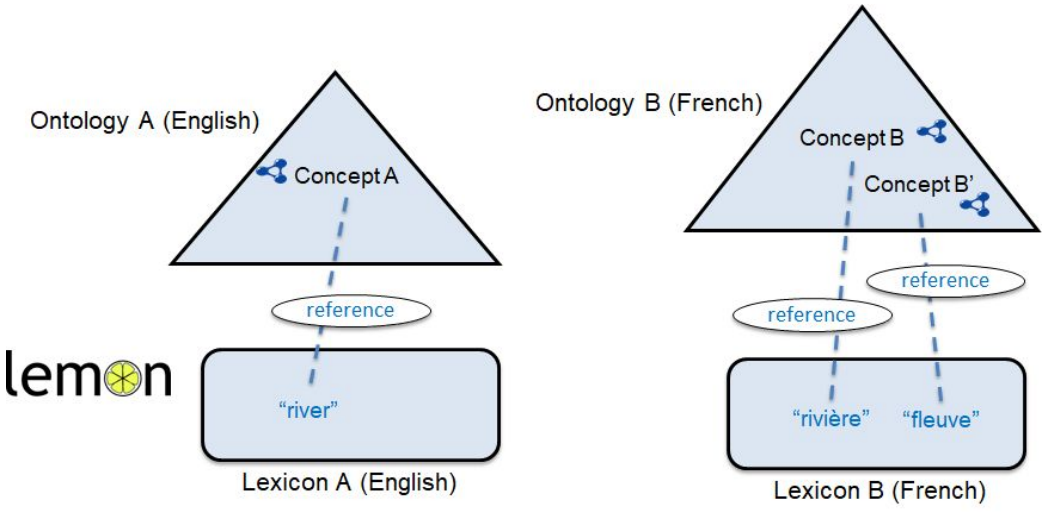


Figure 5: linking between the conceptual and the lexical layers

Level B: linking at the conceptual level. Consider for instance the cross-lingual example in Figure 6, where two ontologies in EN and FR, with their corresponding lexical layers modelled as OntoLex-Lemon lexicons, are put in relation. In the example, the concept that describes “river” in ontology A is linked to the concept that describes “rivière” in Ontology B, by a subsumption relation (**`rdfs:subclassOf`**).

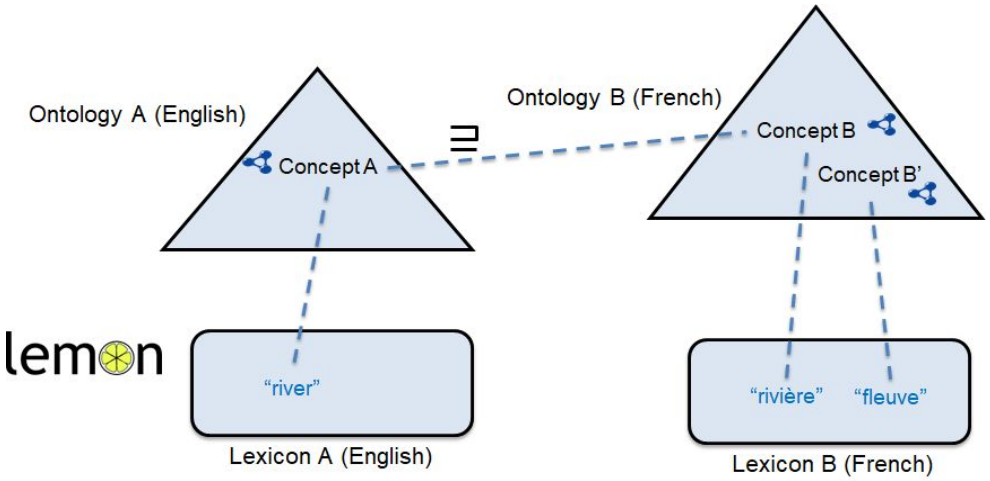


Figure 6: linking at the conceptual level

Level C: Linking at the lexical level. In this case, links are established among the information contained in the lexicons, as in the example pictured in Figure 7, where a translation relation is established between lexical senses.

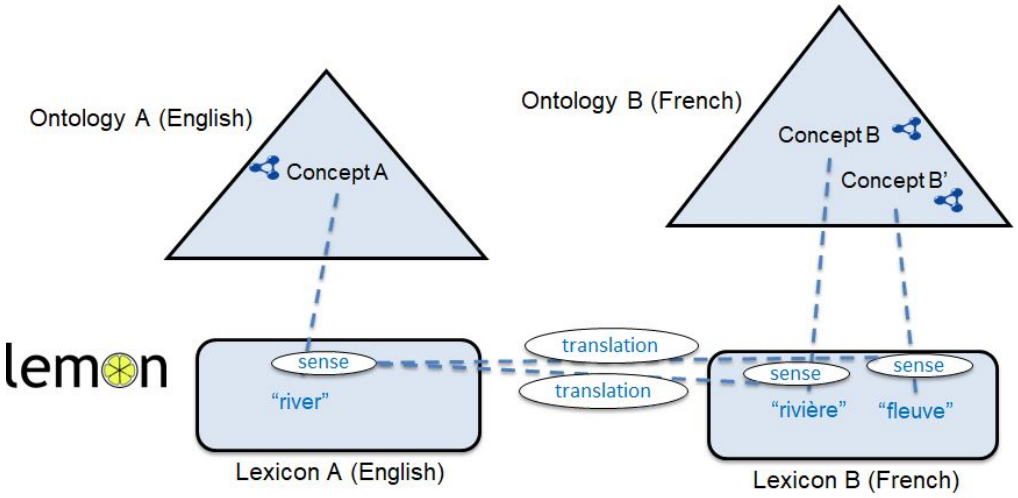


Figure 7: Linking at the lexical level

The discovered links can be **monolingual/cross-lingual** and techniques supporting link discovery can be either **automatic** or **semi-automatic**.

In the following subsections we describe the main techniques that we have started exploring in this early stage of the project.

3.2 Ontology lexicalisation (A)

The goal of ontology lexicalization is to enrich and link existing ontologies with lexical entries that verbalize the ontology elements, ideally across languages. The members of the consortium have been working on the problem of ontology lexicalization previously. Lemonade[13] (Rico and Unger, 2015), for example, is a web application aimed at assisting users to create *lemon lexicalization from an existing ontology*. It is a prototype to show the utility of creating natural language sentences for users while they are providing the parameters of the lexicalization. Any error in the lexicalization produces a wrong sentence, while correct lexicalizations produce correct sentences. The three most used lemon patterns (Class Noun, State Verb and Relational Noun) can be created for any ontology in three languages (English, Spanish and German). VocBench 3 (Stellato et al., 2017) is "a web-based collaborative thesaurus (and ontology) editor supporting access control, history, and structured validation workflows" (Fiorelli et al., 2018). This version of VocBench has a new feature called "custom forms" that is used to create "custom forms for OntoLex-Lemon". This feature is intended to map *lemon patterns* to *custom forms* but, as concluded by the authors, "do not completely satisfy the need for a comprehensive OntoLex-Lemon editor" and this functionality will be addressed in the next editions of the tool.

Regarding the ontology lexicalization research challenge, in Pret-A-LLOD we seek to achieve three main results:

- We aim at extending the functionality of the Lemonade tool to provide users with a quicker way to lexicalize ontologies using *lemon patterns*. Among the new features are included: support for multiple users and UI redesign for a more effective interaction. The software is being refactored to provide a general infrastructure to allow an easy integration of lemon patterns and Grammatical Framework (GF) on top of which new web apps can be created. This general infrastructure will be released as an R package.
- We intend to develop a new concept of "Grammar-as-a-Service" (GaaS) that automatically generates a task-specific grammar from an existing OntoLex-Lemon lexicon. A first prototype is currently being developed and will be described in more detail in future versions of the Research Challenge deliverable. These GaaS will support publication as LLOD resources.
- We intend to develop a framework for instantiating QA systems for a particular ontology on the basis of a question grammar generated by a GaaS. This will reduce the time and effort needed to build QA systems, only requiring a lemon lexicon for a given ontology.
- Algorithms supporting the (semi-) automatic induction of Ontolex-Lemon lexica from corpora that are aligned with the ontology in question, extending previous work (Walter et al. 2016).

[13] http://lemonadetools.linkeddata.es/

In this deliverable we describe our progress in developing algorithms that can automatically extract lexicalization patterns from corpora. A property lexicalization pattern can be applied, for example, in the context of natural language generation from RDF data to natural language text. For example, a pattern can express that an RDF triple such as (**dbr:Syntactic_Structures**, **dbo:author**, **dbr:Noam_Chomsky**) can be verbalized as "Syntactic Structures was written by Noam Chomsky" (given the **rdf:labels** of **dbr:Syntactic_Structures** and **dbr:Noam_Chomsky**).

Due to the variability of natural language there can be many ways in which a relation can be expressed. Thus, we follow a data-driven approach instead of following a manual approach that would be tedious work and might not lead to lexicalization patterns that sound natural. By following a data-driven approach, domain-specific lexicalization patterns can be obtained. Given a set of sentences that all express the same relation, we perform frequent subgraph mining of RDF-graph-based representations of the dependency-parsed sentences to identify what these sentences have in common. Ideally, the only thing that these sentences have in common is that they all express the same relation and the shared pattern can be used to lexicalize a relation.

We are currently developing an approach based on frequent subgraph mining (FSM) to extract lexicalization patterns from dependency parse corpora. Given a set G of graphs and a threshold value $1 \leq \tau \leq |G|$, frequent subgraph mining consists in the task of identifying all graphs that are subgraph to at least *tau* graphs in G.

An RDF graph g_1 is subgraph to another RDF graph g_2 if an injection $\phi : B_{g_1} \rightarrow (B_{g_2} \cup U_{g_2} \cup L_{g_2})$ where ϕ maps each blank node in g_1 to a node in g_2, such that $\phi(g_1) \subseteq g_2$, which means that the set of triples $\phi(g_1)$ is a subset of the set of triples g_2. Frequent subgraph mining is computationally complex - subgraph isomorphism is known to be NP-complete.

So far, we have collected a corpus where sentences are annotated against the DBpedia knowledge base or against the Wikidata knowledge base. We build in particular on the T-rex corpus provided by Elsahar et al. (2018). For each annotated sentence it is known which RDF predicate is expressed, which sequence of tokens express the subject and which sequence of token express the object. Furthermore, the KB identifiers of subject and object are known. We apply the graph mining algorithm per property to groups of sentences that contains the same verb lemma, noun lemma or adjective lemma and that have at least 5 elements. The evaluation and fine-tuning of the algorithm and thresholds is ongoing and will be reported in future version of the research challenge deliverable (v2).

3.3 Ontology matching (level B)

The target of this activity is to develop a general purpose cross-lingual ontology matching tool. Such a tool will be “general” in the sense that it will be domain neutral but easily adaptable to the requirements of the project’s pilots. It will operate with any two input ontologies given in standard formats (OWL, RDFS, …) and produce a resulting alignment (in the Alignment Format and other suitable format).

To that end, we take as starting point CIDER-CL (Gracia et al., 2013), a monolingual and cross-lingual ontology matching system developed by members of the Prêt-à-LLOD consortium in the past. A series of adaptations to CDER-CL have been started to allow the system to improve their cross-lingual capabilities, primarily through the exploration of the use of monolingual and multilingual word embeddings (Ruder et al., 2019), knowledge embeddings (Cai et al., 2017), as well as other related techniques based on distributional semantics.

We plan to compare the resulting system with other state-of-the art cross-lingual ontology matching tools, based on the "Multifarm" track of the Ontology Alignment Evaluation Initiative (OAEI) (Algergawy et al., 2018).

3.4 Translation Inference Across Dictionaries (level C)

Since the "translation" relation is core in this project, the initial efforts for connecting data at the lexical level have been devoted to such a type of links. Particularly, we focused on a scenario in which a number of pre-existing translations is available among different dictionary data in different languages (e.g., as in the Apertium RDF graph (Gracia et al., 2018) and we want to find translations between data in two languages for which there is no available translations.

Our effort has been two-fold: (1) we have organised a shared task for the controlled evaluation of such a type of techniques, i.e. the "Translation Inference Across Dictionaries (TIAD 2019) shared task" (see https://tiad2019.unizar.es/), which comprises two baselines and a blind dataset that is used as golden standard; and (2) we are developing a system aimed at outperforming such baselines, based on a combination of the One Time Inverse Consultation (OTIC) algorithm (Tanaka and Umemura, 1994) and the exploration of the graph density (Villegas et al., 2016).

The objective of the TIAD shared task (Gracia et al., 2019) was to explore and compare methods and techniques that infer translations indirectly between language pairs, based on other bilingual resources. Such techniques will help in auto-generating new bilingual and multilingual dictionaries based on existing ones. Three contributing systems that participated at TIAD 2019 were developed by Prêt-à-LLOD partners.

An initial round of experiments with results sent by TIAD 2019 participants is reported in https://tiad2019.unizar.es/results.html. Currently, new experiments based on our new system are in progress and will be reported in the next version of this document.

3.5 Linking of lexical data (level C)

A version of the generic ontology matching system described in section 3.2 will be developed to operate with a particular type of data that is core in this project, that is with lexical data (e.g., data coming from dictionaries, or from lexicalised ontologies), taking into account the particular requirements of the pilots. For instance to compare dictionary data at

the sense level as it is planned in Pilot II, dedicated to linking lexical knowledge in order to facilitate wider application of lexicographic resources (dictionaries of different types, thesauri, etc.) in language technology areas such as multilingual search, cross-lingual document retrieval, domain adaptation, and lexical translation.

The resulting component will be tested with data samples created for a previous effort on cross-dictionary sense linking (Saurí et al., 2019) at Oxford University Press, the Prêt-à-LLOD partner in charge of Pilot II. That initial experiment was carried out without benefiting from the data processing and management functionalities contributed by the Linked Data paradigm.

3.6 Integration in a common framework: NAISC

Naisc[14] is a tool for data linking that has been developed at NUIG[15] in the context of the Insight Centre and the ELEXIS H2020 project and it is intended that the results of data linking in Prêt-à-LLOD will be integrated into this work. Naisc views the process of data linking as a sequence of tasks that can be performed in many ways and by changing components in this data linking tasks can be customized to a particular task. The pipeline of Naisc is as follows:

1. **Analysis**: The two datasets to be used are analyzed for the properties and potential entities that exist in the dataset. For example, it is checked if the datasets conform to any existing standards such as OntoLex-Lemon, SKOS or OWL
2. **Blocking**: A rough algorithm is used to select candidates that may match, for example because they contain labels that overlap in some of the words or character n-grams. The base strategy here is to match all elements in one dataset with those in another, however this can lead to a very large number of matches and hence long computation times, so other strategies are generally applied
3. **Prelinking**: As a result of the blocking, normally some entities can already be matched without further analysis. Typically this is due to two entities unambiguously having exactly the same label.
4. **Lens**: A set of string representation of each entity is extracted so that string similarity techniques may be applied
5. **Text Features**: A variety of string similarity entities are applied to the extracted strings to estimate similarity
6. **Graph Features:** In addition to NLP analysis, other features based on the graphs of the two datasets are extracted.
7. **Scoring**: All features are combined into a single score representing the likelihood of two elements from each dataset being linked
8. **Matching:** In the final step, the matching across all elements of the two datasets is considered holistically. In this step, **constraints**, are applied for example disallowing or penalizing multiple links from a single entity in a dataset.

Naisc is available as an open source toolkit from https://github.com/insight-centre/naisc/

[14] Irish for ‘links’, pronounced as ‘nashk’.
[15] The Prêt-à-LLOD project partner National University of Ireland Galway

3.7 Prospective software deliverable(s) and data sets

The software component associated to that task, i.e., the *Prêt-à-LLOD Linking* technical component, will be delivered on M24. Figure 8 gives an overview of the components that will take part in *Prêt-à-LLOD Linking* and how they will interact.

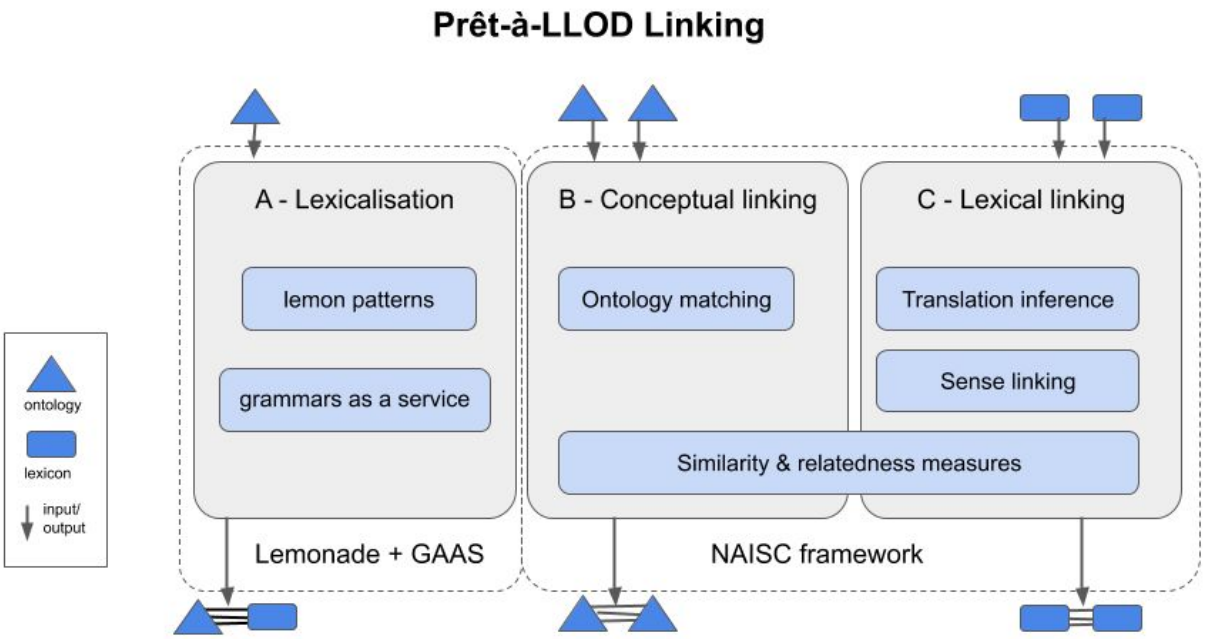


Figure 8: an Overview of the Prêt-à-LLOD Linking component

We devise at least the following services that the *Prêt-à-LLOD Linking* technical component will support:

Service: Extension to **lemonade**
Responsible: UPM (Universidad Politécnica de Madrid)
Input: ontology and lexicalisation data (manually provided)
Output: lemon pattern instances
Description: In the same way that Lemonade (Rico and Unger, 2015) produced lemon pattern instances, this new tool will produce the same data but in a quicker way. This is a key feature when dealing with big ontologies like Wikipedia.

Service: **Ontology lexicalisation**
Responsible: UNIBI (Universität Bielefeld)
Input: linguistic resources (including corpora)
Output: OntoLex-Lemon lexica, RDF-graph based patterns/SPARQL queries
Description: Algorithm for inducing Ontolex-lexicalizations for a given ontology on the basis of a given corpus

Service: **Grammar-as-a-service**
Responsible: UNIBI
Input: OntoLex-Lemon lexicon, ontology (in OWL), knowledge base (in RDF)
Output: A question answering grammar that can be used as the basis to develop a QA system
Description: A tool that generates grammars as a service on the basis of an OntoLex-Lemon lexicon and a given ontology

Service: **Question Answering System as a service**
Responsible: UNIBI
Input: a question answering grammar generated by the GaaS service (see above)
Output: Web application
Description: A framework that allows to instantiate an incremental QA system on the basis of a GaaS that configures the QA systems

Service: semantic **similarity** between ontology entities
Responsible: UNIZAR (Universidad de Zaragoza)
Input: two ontology entities
Output: semantic similarity value in [0,1]
Description: Computation of the degree of similarity between two ontology entities documented in the same or different languages

Service: semantic **relatedness** between ontology entities
Responsible: UNIZAR
Input: two ontology entities
Output: semantic relatedness value in [0,1]
Description: Computation of the degree of relatedness between two ontology entities documented in the same or different languages

Service: Generic **ontology matching**
Responsible: UNIZAR
Input: two monolingual or multilingual ontologies
Output: an alignment in the Alignment Format
Description: Generic ontology matching service for the discovery of cross-lingual and monolingual semantic equivalences between classes and properties of the two ontologies.

Service: **lexicon matching**
Responsible: UNIZAR
Input: two lexicons in the same or different languages
Output: a set of ontolex-based correspondences
Description: service for the discovery of links across OntoLex-Lemon lexicons

Service: **translation inference**
Responsible: UNIZAR
Input: two dictionaries
Output: a set of translations
Description: service for the discovery of indirect translations across two initially disconnected dictionaries that belong to the same RDF graph of dictionary data.

Service: **imprecise/vague translation inference**
Responsible: UNIZAR
Input: two dictionaries annotated with degrees of truth
Output: a set of translations annotated with degrees of truth
Description: service for the discovery of indirect translations across two initially disconnected dictionaries that belong to the same RDF graph of dictionary data. Input dictionaries must be annotated with degrees of truth denoting imprecision/vagueness, i.e., each translation can be annotated with a degree in [0, 1] estimating to which extent a translation holds.

Service: **uncertain translation inference**
Responsible: UNIZAR
Input: two dictionaries annotated with degrees of certainty
Output: a set of translations annotated with degrees of certainty
Description: service for the discovery of indirect translations across two initially disconnected dictionaries that belong to the same RDF graph of dictionary data. Input dictionaries must be annotated with degrees of certainty denoting uncertainty, i.e., each translation can be annotated with a degree in [0, 1] measuring our confidence in the correctness of the translation.

4. Workflows for Portable and Scalable Semantic Language Services [T3.3]

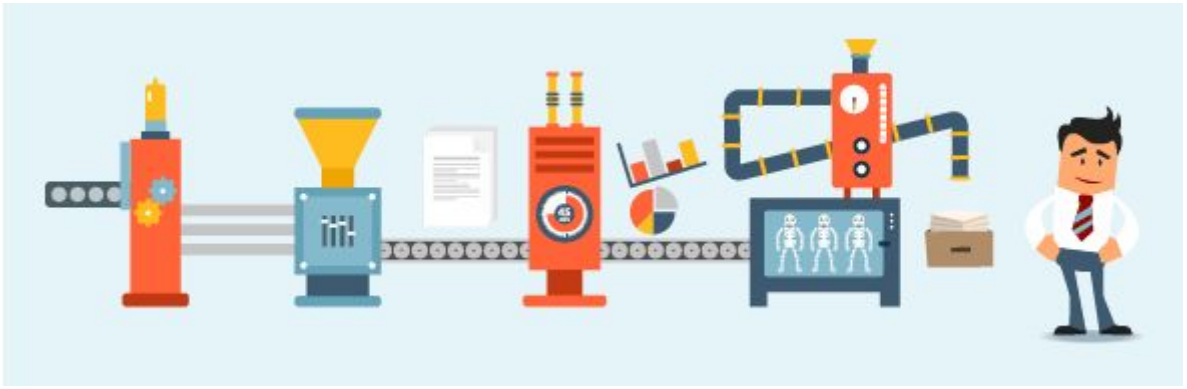
The NLP workflows system will be realized by a platform called Teanga, which is a linked data based platform for natural language processing (NLP). This deliverable will describe the current state of the system, and the plans for development over the course of the project.

Teanga enables the use of many NLP services from a single interface, whether the need was to use a single service or multiple services in a pipeline. Teanga focuses on the problem of NLP services interoperability by using linked data to define the types of service input and output.

4.1 Concept

Researchers looking to complete NLP tasks use multiple NLP services, but since different providers developed these services, the researchers need to run their data through the first service and then handle the output manually to be added to another service as input.
Indeed, services from different providers seem to be losing the alignment when they are put in a pipeline, and thus losing the interoperability among them.

The concept behind Teanga was to enable the researchers of using a platform that creates this interoperability automatically by aligning services according to the data types from their input and/or output.

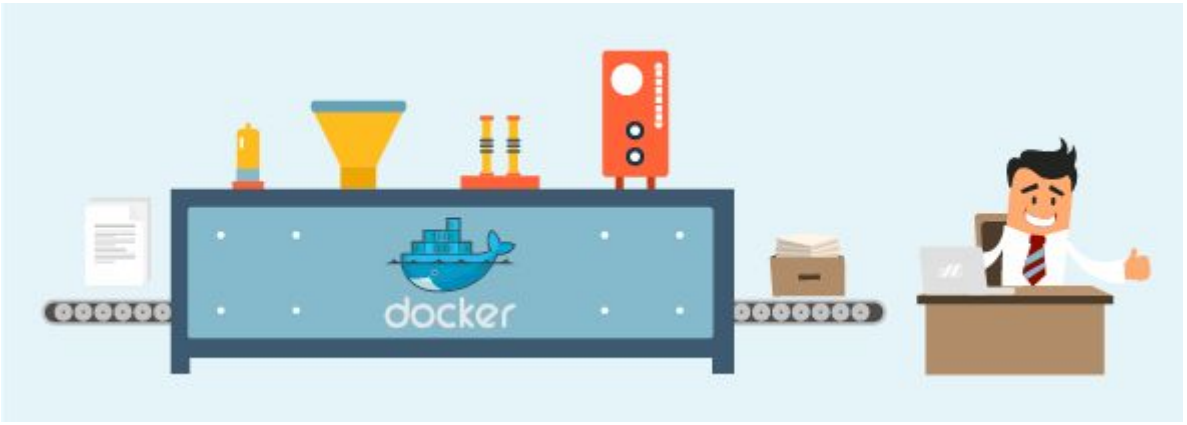


In this way, researchers can use a single interface, and the system would know how to handle passing the data among the services.

In this stage, the first version of Teanga was created. Teanga then could add services from the web or the user's server, but it needed some extra work to install the services and managing them. Add to that, some services are quite hard to handle and install, and they provide different output formats like XML or JSON.

The next step was to let Teanga handle adding the services to the user's device, run them before the user starts using them, and guarantees a comprehensive output format, which is JSON-LD.

To handle these points, we utilised Docker and containerisation for the installation and handling of the services and implemented multiple technologies (Linked Data, OpenAPI) to produce JSON-LD files as the output.



The second version of Teanga should be using Docker and other state-of-the-art technologies to create a black box, where the user needs to download and install the Docker image on their system, and it would handle the rest.

4.2 History

- Teanga started as a Master's thesis[16] for Housam Ziad[17].
- Teanga was first introduced during LREC 2018[18].
- Teanga was originally developed under Science Foundation Ireland Grant for the Insight Centre for Data Analytics (SFI/12/RC/2289).
- In Prêt-à-LLOD the focus will be on making the system scalable and better exploit semantic metadata.

4.3 Teanga Technologies

A key enabling factor for the Teanga platform is the use of the best technologies that exist, in order to enable the service to work efficiently at scale. In particular, we make use of the MEAN stack, which contains open-source tools as follows:

1. Easy-to-use interface by using the Bootstrap 4 library.
2. Stability and maintenance of the Web framework by using the Angular 7 library to build the front-end.
3. Using the NodeJS library to run the server and the back-end parts.
4. MongoDB is used for data storage, as it uses a JSON-like data structure, which corresponds to our use of JSON-LD files.
5. Using Docker as containerization technology so that the user can download and run Teanga in a simple process of only one step.
6. JSON-LD files for input and output, and to create an interoperable model among the services.

4.4 Related Work

In the domain of NLP architecture, multiple frameworks, toolkits, and suites have been created, and each of them uses a different approach to creating interoperability among all their services, and, by that, reduce the amount of manual work needed to process data. Among these are the LAPPS Grid (Ide et al., 2015) and its Galaxy front-end (Ide et al., 2016), GernEdiT: A Graphical Tool for GermaNet Development (Henrich and Hinrichs, 2010),
Language Grid: An Infrastructure for Intercultural Collaboration (Ishida, 2006), and Unstructured Information Management Architecture (UIMA) (Ferrucci and Lally,
2004).

Some problem with the applications of other platforms is that some of them only run on a desktop machine or rely on a platform-specific program, e.g. Eclipse plugins. For example, in the case of UIMA, it's only a middleware architecture to be taken into account while

[16] https://search.library.nuigalway.ie/permalink/f/1pmb9lf/353GAL_SP_DSNUIG-28-689
[17] http://hmz.ie
[18] http://www.lrec-conf.org/proceedings/lrec2018/summaries/106.html (Ziad et al., 2018)

developing a new NLP tool. For example, it doesn't provide the user with an interface to process data. UIMA also is like GATE when it comes to the complexity of installing and setting up the environment to be used in the development process. It's intended for an expert who is developing an NLP tool and wants to include it in the UIMA environment.

The other platforms do not sufficiently consider user experience, as user-experience problems can be seen in two parts. The first is installing and running them for the first time, which, for all of them, requires a high level of expertise in a specific environment or programming language, and for some of them, is a time-consuming process. The other part can be seen in the user interface, as some of them don't include a graphical user interface, and users need to run commands from the terminal. Others like GATE created an interface but, even to its developers, (Cunningham, 2002): "The visual interface is complex and somewhat non-standard.". While in the case of LAPPS, their interface seems to be hard to be used by an unskilled user.

A common issue of all of these platforms is the fact that developers have to follow specific standards or guidelines while developing their services before they can be added to the framework to guarantee the interoperability of the platform. While a recent project, OpenMinTeD, is working on the standardisation of tools for NLP, these proposals have yet to bear fruit. With Teanga, developers of already existing services can add their services to the platform only by including a configuration file in the container.

4.5 Linked Data in Teanga

The use of Linked Data and Semantic Web technologies in applications delivers structured information, which can be used and queried by a flexible and extensible way to get a better understanding of the data. In particular, the platform exploits the Semantic Web model of data types, to describe the possible format of input and output to services. Since Teanga is designed to deal with any NLP service, and since we can't predict all possible datatypes that may be used by NLP services, we use Semantic Web technologies to define the data types that pass through the services. By doing this, we can let the machine running Teanga understand what data it is processing and how to handle moving it around all the services in a pipeline. The use of Semantic Web technologies will help Teanga, as a platform, to understand the data input and output for each of the services added to the system, and will contribute to creating data interoperability among services to create clear and straightforward pipelines when the user needs to use them, as URIs can be dereferenced to find more info.

In particular, there have been a number of models for the representation of linguistic structures used in natural language processing as linked data. The major type of data handled by Teanga is corpus data, and there are a number of models for stand-off annotation of corpora data that have been developed including the NLP Interchange Format (Hellmann et al., 2013) and the Open Annotation format (Sanderson et al., 2013).

In addition, more detailed linguistic models such as POWLA (Chiarcos, 2012) as well as specific models such as for parse trees (Chiarcos and Fàth, 2017). In addition, we rely on

common models for linguistic categories such as those proposed by ISOcat (Windhouwer and Wright, 2012), now maintained by the CLARIN Concept Registry (Schuurman et al., 2016), or open repositories such as LexInfo (Cimiano et al., 2011) and OLiA (Chiarcos and Sukhareva, 2015). Finally, we can also use models for representing lexical information on the Web, in particular, the Lexicon Model for Ontologies (McCrae et al., 2012; Cimiano et al., 2016).

As an example, for a machine translation service, we would find that both the source language and target language share the same type because both are referring to a natural language. In this case, we can use an existing type such as **`Language`** from the Lexvo Ontology (de Melo, 2015) and require that values are given as one of the known inputs to this service. We can use JSON-LD aliases to simplify this creating a mapping between the string, e.g., `en`, and the URL, e.g., **`http://www.lexvo.org/page/iso639-3/eng`**). Moreover, for other datatypes such as strings, we can reuse other standards such as XML Schema to define basic datatypes (such as **`xsd:string`**) or using custom datatypes that can be defined using the OWL vocabulary (McGuinness, 2004).

This can be used to join services, for example, if we have a sentiment analysis service that accepts multilingual input and /or output, this service would have an input to enter the text, and an option to select the text language. In this case, the language in the sentiment analysis is of the same type as the languages in the machine translation. If we want to pass data from the sentiment analysis to the machine translation, we can have something as shown in Figure 9.

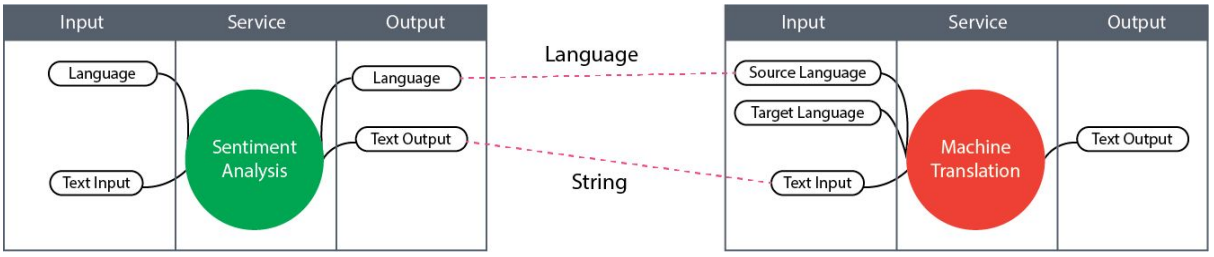


Figure 9: Showing how services share the same types, and how to connect a simple pipeline.

As many of the datatypes used by NLP services are basic or common values such as plain text or language, linked data methods can help the machine understand what the type of this piece of data is and where to connect that data once we have a pipeline of services. Furthermore, as each of these types is mapped to a URL it is possible to find extra information about it, such as a description, by dereferencing the URL and to provide restrictions, backed by description logic to detect inconsistent combinations of services.

Housam Ziad's thesis described above gives better details on the description of how Teanga was developed at first.

4.6 Teanga's Ontology

Teanga contains an ontology to describe details about services inputs and outputs. This ontology is meant to be built upon existing NLP services ontologies and then filling the gaps

if any. The system uses the ontology to deeply understand the datatypes on each service added to the platform.

4.7 How Teanga works?

Figure 10 shows the two main parts of how Teanga works.

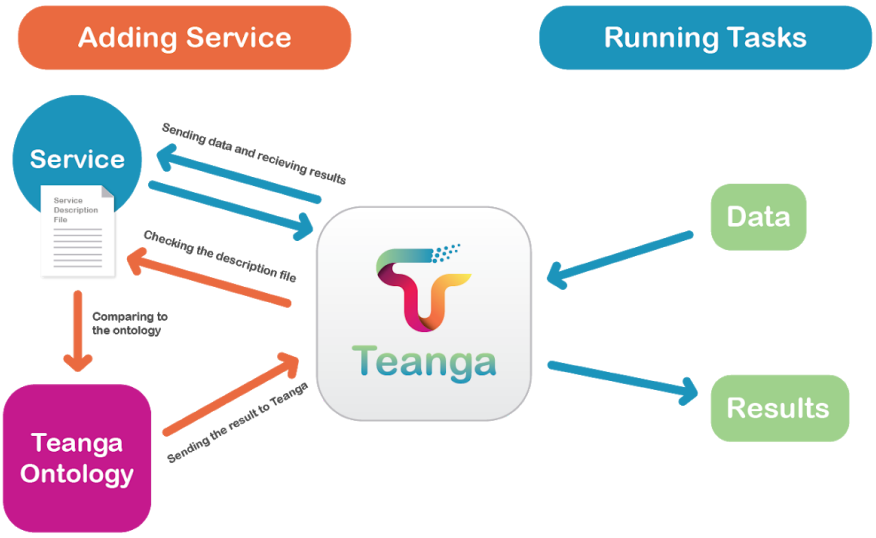


Figure 10: Managing services and tasks in Teanga

Adding a Service, The Orange Line:
If a user is trying to add a service to Teanga, the platform checks for a service description file. This file holds information about the service. Teanga checks the file and compares the entries to the ontology, and then add that service to the services list.

Running a Task, The Blue Line:
When a user creates a workflow of NLP services pipeline and uploads their data, the platform would serve as an orchestration tool among the services in the pipeline to deliver the final results in JSON-LD format.

4.8 Current Status

Currently, we have two options to deploy Teanga:

1. Case 1 - One combined app: We run only the Node REST API, which renders the Angular App too. We need a server that runs the NodeJS code and at the same time delivers the Angular code.
2. Case 2 - Two separate apps: In this mode, we deploy the front-end Angular app alone, and deploy the NodeJS REST API (Node, Express, and MongoDB).
   With this setup, the front-end needs a static host which only serves HTML, CSS, and JavaScript code, while the back-end needs a server that runs the NodeJS code.

The development code is using the two separate apps option, and if Case 1 is needed, the project needs to be restructured easily to combine everything in one app. Please check section 6.3 Building a Single App for more information on how to.

The MEAN stack parts are being used as follows:

Back-end

The back end is written using NodeJS, and the routing is being done using ExpressJS library. While Node works as a back-end server, ExpressJS is needed to facilitate the routing easily without wasting time and effort to make these with Node code. The back-end is working as a RESTful API with different endpoints to execute the back-end commands needed for the platform to run.

Front-end

Teanga's front-end depends on Angular version 7, meaning it's built using TypeScript[19], an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for development of large applications and transcompiles to JavaScript.

OpenAPI support

Any service in Teanga needs to have a description file; this file includes information about the service, especially its input and output. After testing our approach with creating JSON-LD files to describe a service, we found out that the OpenAPI specifications are mature and well designed to be used with any API-like services. By using OpenAPI, we save time and avoid errors when writing code.

4.9 Ongoing & Future Work

Future work includes two parts:

Research

1. Integrating with LAPPS Grid: The Language Application Grid (Ide et al., 2015) is an open, interoperable web service platform for natural language processing (NLP) research and development. It uses LAPPS Web Service Exchange Vocabulary, which defines an ontology of terms for a core of linguistic objects and features exchanged among NLP tools that consume and produce linguistically annotated data. We will bring the LAPPS Web Service Exchange Vocabulary to Teanga and build Teanga's Ontology based on it. Furthermore, LAPPS Grid provides many NLP services like serialization, metadata, and annotations, which will be added to Teanga's list of services.

[19] https://www.typescriptlang.org/

2. Conducting research on different Ontologies and trying to add them to Teanga, the proposed ontologies are DKPro[20], OpenMinted[21], and MetaShare[22].

Engineering

As a part of enhancing the user experience, the following features are being added to Teanga:

1. The predefined tasks, which cover the common experiments that are used in NLP research, we put in a list to choose from, which the system will use to create the whole workflow, saving multiple steps for the user. For example, when the user selects suggestion mining on multilingual text, the system will place the machine translation and the suggestion mining services in the graph, connect them, and then the user only has to choose the languages for the machine translation service.
2. The ability to save and load a saved workflow, in case the user needs to rerun the same experiment in future on a different data set. They can upload the data and just load the workflow in one click.
3. Automatic source language detection.
4. Adding related information to the results page, e.g. how long the experiment took.
5. Adding a service that can handle multiple types of inputs like different types of files, web links, and zip files.
6. Better error handling after testing Teanga with more services.
7. Enhancing the Docker control to be limited to Teanga only images and containers.

Others

1. Defining if two services can't be in a pipeline.
2. Adding the ability to write some JavaScript code to handle alignment for big data if the system couldn't align automatically.
3. Enabling the user to do alignment manually if the system failed to do that automatically.

[20] https://dkpro.github.io/
[21] http://openminted.eu/
[22] http://www.meta-share.org/

5. Summary

The focus of WP3 is in the development of usable tools that will allow data producers to make their resources ready-to-use and to allow data consumers to easily apply this data into their workflows. To this end, we are looking at the development of a number of tools: firstly the **FINTAN** (Flexible and Integrative Transformation and Annotation eNgineering) tool will be developed that will make an easy and generic way to convert data into RDF based on the usage of SPARQL update queries, it is intended that this would make the process significantly easier and more generic. We have developed a number of case studies of formats across a diverse spectrum of language data that we intend to apply this to. Secondly, a key goal of the project is to allow multiple datasets to be used in concert with one another and this requires that links be established between the resources. We have divided this task into two tools, firstly a specific tool for linking ontologies and lexicons, which will be based on previous work on **M-ATOLL**. Secondly, we will look at ontology and dataset alignment through new modules to be integrated into the **Naisc** system, which will give a general-purpose tool that adapts itself to a wide range of linking tasks. Finally, we are furthering the development of the **Teanga** framework, which allows NLP pipelines to be built using linked data and we are intending to focus on extending this framework to be scalable to deployment in grids of hundreds of machines, hence allowing linked data to enable the integration of sophisticated pipelines.

References

Algergawy, A. et al. (2018). Results of the Ontology Alignment Evaluation Initiative 2018. In *Proceedings of the 13th International Workshop on Ontology Matching (OM 2018) at the 17th International Semantic Web Conference, ISWC 2018*. Monterey, USA.

Bond, F., and Foster, R. (2013). Linking and Extending an Open Multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.* Sofia, pp. 1352–1362. URL http://aclweb.org/anthology/P13-1133.

Bond, F., and Paik, K. (2012). A survey of wordnets and their licenses. In *Proceedings of the 6th Global WordNet Conference, GWC 2012*. Matsue, Japan, pp. 64–71.

Bond, F., Vossen, P., McCrae, J.P. and Fellbaum, C. (2016). CILI: the Collaborative Interlingual Index. In *Proceedings of the 8th Global WordNet Conference, GWC 2016*. Bucharest, Romania.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2017). A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. In *IEEE Transactions on Knowledge and Data Engineering*, 30(9), pp. 1616–1637.

Chiarcos, C. (2012). POWLA: Modeling linguistic corpora in OWL/DL. In *Proceedings of the Extended Semantic Web Conference, ESWC 2012*, pp. 225–239.

Chiarcos, C., Donandt, K., Ionov, M., Rind-Pawlowski, M., Sargsian, H., Wichers-Schreur, J., Fäth, C. (2018a). Universal Morphologies for the Caucasus region. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

Chiarcos, C., Donandt, K., Sargsian, H., Wichers-Schreur, J., and Ionov, M. (2018b). Towards LLOD-based Language Contact Studies: A Case Study in Interoperability. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

Chiarcos, C., and Fäth, C. (2017). CoNLL-RDF: Linked Corpora Done in an NLP-Friendly Way. In *Language, Data, and Knowledge, LDK 2017.* Galway, Ireland, pp. 74–88.

Chiarcos, C. and Sukhareva, M. (2015). OLiA–ontologies of linguistic annotation. *Semantic Web*, 6(4), pp. 379–386.

Cimiano, P., Buitelaar, P., McCrae, J., and Sintek, M. (2011). LexInfo: A declarative model for the lexicon-ontology interface. In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1), pp. 29–51.

Cimiano, P., Chiarcos, C., McCrae, J.P., Gracia, J. (2019). Linguistic Linked Data - Representation, Generation and Applications. Springer. [TO APPEAR]

Cimiano, P., McCrae, J., Rodriguez-Doncel, V., Gornostaya, T., Gomez-Perez, P., Benjamin Siemoneit, P. and Lagzdins, P. (2015). Linked Terminology: Applying Linked Data Principles to Terminological Resources. In *Proceedings of the eLex 2015 conference*. Herstmonceux Castle, United Kingdom.

Cimiano, P., McCrae, J.P., and Buitelaar, P. (2016). Lexicon Model for Ontologies: Community Report. URL https://www.w3.org/2016/05/ontolex/

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gomez-Perez, A., Garcia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., and Wunner, T. (2012). Interchanging Lexical Resources on the Semantic Web. In *Language Resources and Evaluation*, 46(4), pp. 701–719.

Das, S., Sundara, S., Cyganiak, R. (2012). R2RML: RDB to RDF mapping language. W3C recommendation, World Wide Web Consortium. URL https://www.w3.org/TR/csv2rdf/

Declerck, T., and Racioppa, S. (2019). Porting Multilingual Morphological Resources to OntoLex-Lemon. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*. Varna, Bulgaria.

Elsahar, Hady, Vougiouklis, Pavlos, Remaci, Arslen, Gravier, Christophe, Hare, Jonathon, Simperl, Elena and Laforest, Frederique (2018). T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

Fellbaum, C. (ed.) (1998). WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Fiorelli, M., Stellat, A., Lorenzetti, T., Turbati, A., Schmitz, P., Francesconi, E., and Batouche, B. (2018). Towards OntoLex-Lemon editing in VocBench 3. In *AIDAInformazioni*, 36(special issue), pp. 81–102.

Gonzalez-Agirre, A., Laparra, E., and Rigau, G. (2012). Multilingual Central Repository version 3.0: upgrading a very large lexical knowledge base. In *Proceedings of the 6th Global WordNet Conference, GWC 2012*. Matsue, Japan.

Gracia, J., and Asooja, K. (2013). Monolingual and cross-lingual ontology matching with CIDER-CL: Evaluation report for OAEI 2013. In *Proceedings of the 8th Ontology Matching Workshop (OM 2013), at 12th International Semantic Web Conference, ISWC 2013*. Sydney, Australia.

Gracia, J., Kasabi, B., and Kernerman, I. (eds.) (2019): Proceedings of TIAD-2019 Shared Task – Translation Inference Across Dictionaries, co-located with the 2nd Conference on Language, Data and Knowledge, LDK 2019. Leipzig, Germany. [TO APPEAR]

Gracia, J., Villegas, M., Gómez-Pérez, A., and Bel, N. (2018) The Apertium bilingual dictionaries on the web of data. In *Semantic Web*, 9(2), pp. 231–240.

McGuinness, D. L., Van Harmelen, F., et al. (2004). OWL Web Ontology Language Overview. W3C recommendation, World Wide Web Consortium.

Henrich, V., and Hinrichs, E. (2010). GernEdiT-the GermaNet editing tool. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics - System Demonstrations*. Uppsala, Sweden.

Hellmann, S., Lehmann, J., Auer, S., and Brümmer, M. (2013). Integrating NLP using linked data. In *Proceedings of the 12th International Semantic Web Conference, ISWC 2013*. Sydney, Australia, pp. 98–113.

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., DiPersio, D., Shi, C., Suderman, K., Verhagen, M., Wang, D., and Wright, J. (2015). The Language Application Grid. In *International Workshop on Worldwide Language Service Infrastructure*, Kyoto, Japan, pp. 51–70.

Ishida, Toru. Language grid: An infrastructure for intercultural collaboration. In Proceedings of the *International Symposium on Applications and the Internet, SAINT 2006*. Phoenix, Arizona, USA.

Lösch, A., Mapelli, V., Piperidis, S., Vasiļjevs, A., Smal, L., Declerck, T., Schnur, E., Choukri, K., and van Gen-abith, J. (2018). European Language Resource Coordination: Collecting Language Resources for Public Sector Multilingual Information Management. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

de Melo, G. (2015). Lexvo.org: Language-related information for the linguistic linked data cloud. *Semantic Web*, 6(4), pp. 393–400.

Morato, J., Marzal, M., Lloréns, J., and Moreiro, J. (2004). WordNet Applications. In *Proceedings of the 2nd Global WordNet Conference, GWC 2004*. Brno, Czech Republic, pp. 270–278. URL http://www.fi.muni.cz/gwc2004/proc/105.pdf.

Petitpierre, D., and Russell, G. (1995). MMORPH: The Multext Morphology Program. Multext deliverable 2.3.1, ISSCO, University of Geneva.
URL http://www.issco.unige.ch/downloads/multext/mmorph.doc.ps.tar.gz.

Pianta, E., Bentivogli, L., and Girardi, C. (2002). MultiWordNet: Developing an Aligned Multilingual Database. In *Proceedings of the 1st International Conference of the Global Wordnet Association*. Mysore, India, pp. 293–302.

Rico, M., and Unger, C. (2015). Lemonade: A web assistant for creating and debugging ontology lexica. In Proceedings of the *International Conference on Applications of Natural Language to Information Systems, NLDB 2015*. Passau, Germany. Lecture Notes in Computer Science, 9103. pp. 448-452. Springer, Cham.

Ruder, S., Vulić, I., and Søgaard, A. (2019) A Survey Of Cross-lingual Word Embedding Models. In *Journal of Artificial Intelligence Research*.

Sagot, B., and Fišer, D. (2008). Building a free French wordnet from multilingual resources. In *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*. Marrakech, Morocco.

Sanderson, R., Ciccarese, P., Van de Sompel, H., Bradshaw, S., Brickley, D., Castro, L. J. G., Clark, T., Cole, T., De- senne, P., Gerber, A., et al. (2013). Open Annotation data model. W3C community draft, World Wide Web Consortium.

Saurí, R., Mahon, L., Russo, I., and Bitinis, M. (2019). Cross-Dictionary Linking at Sense Level with a Double-Layer Classifier. In *Proceedings of the 2nd Conference on Language, Knowledge and Data, LDK 2019*. Leipzig, Germany.

Schuurman, I., Windhouwer, M., Ohren, O., and Daniel, Z. (2016). CLARIN concept registry: the new semantic registry. In *Selected Papers from the CLARIN Annual Conference 2015*. Wroclaw, Poland, pp. 62–70.

Stellato, A., Turbati, A., Fiorelli, M., Lorenzetti, T., Costetchi, E., Laaboudi, C., ... and Keizer, J. (2017). Towards VocBench 3: pushing collaborative development of thesauri and ontologies further beyond. In *17th European Networked Knowledge Organization Systems Workshop, NKOS 2017*. Thessaloniki, Greece, pp. 39-52.

Sylak-Glassman, J., Kirov, C., Yarowsky, D., and Que, R. (2015). A Language-Independent Feature Schema for Inflectional Morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China, pp. 674–680.
URL https://doi.org/10.3115/v1/P15-2111

Tanaka, K., and Umemura, K. (1994). Construction of a Bilingual Dictionary Intermediated by a Third Language. In *15th International Conference on Computational Linguistics, COLING 1994*. Kyoto, Japan, pp. 297–303.

Tandy, J., Herman, I., Kellogg, G., et al. (2015). Generating RDF from Tabular Data on the Web. W3C recommendation, World Wide Web Consortium.
URL https://www.w3.org/TR/csv2rdf/

Toral, A., Bracale, S., Monachini, M., and Soria, C. (2010). Rejuvenating the Italian WordNet: upgrading, standardising, extending. In *Proceedings of the 5th International Conference of the Global WordNet Association, GWC 2010*. Mumbai, India.

Villegas, M., Melero, M., Bel, N., and Gracia, J. (2016). Leveraging RDF Graphs for Crossing Multiple Bilingual Dictionaries. In *Proceedings of the 10th Language Resources and Evaluation Conference*, LREC 2016, Portorož, Slovenia, pp. 868–876.

Walter, S., Unger, C., and Cimiano, P. (2014). ATOLL - A framework for the automatic induction of ontology lexica. In *Data & Knowledge Engineering*. 94(B), pp. 148-162.

Windhouwer, M. and Wright, S. E. (2012). Linking to linguistic data categories in ISOcat. In *Linked Data in Linguistics, LDL 2012*. Frankfurt, Germany, pp. 99–107.

Ziad, H., McCrae, J., and Buitelaar, P., (2018). Teanga: A Linked Data based platform for Natural Language Processing. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

Appendix: Mapping TBX to OntoLex-Lemon

We briefly describe the TBX Data Model abstracting from the XML specifics in what follows. Figure 11 summarizes the TBX Data Model as a UML diagram.

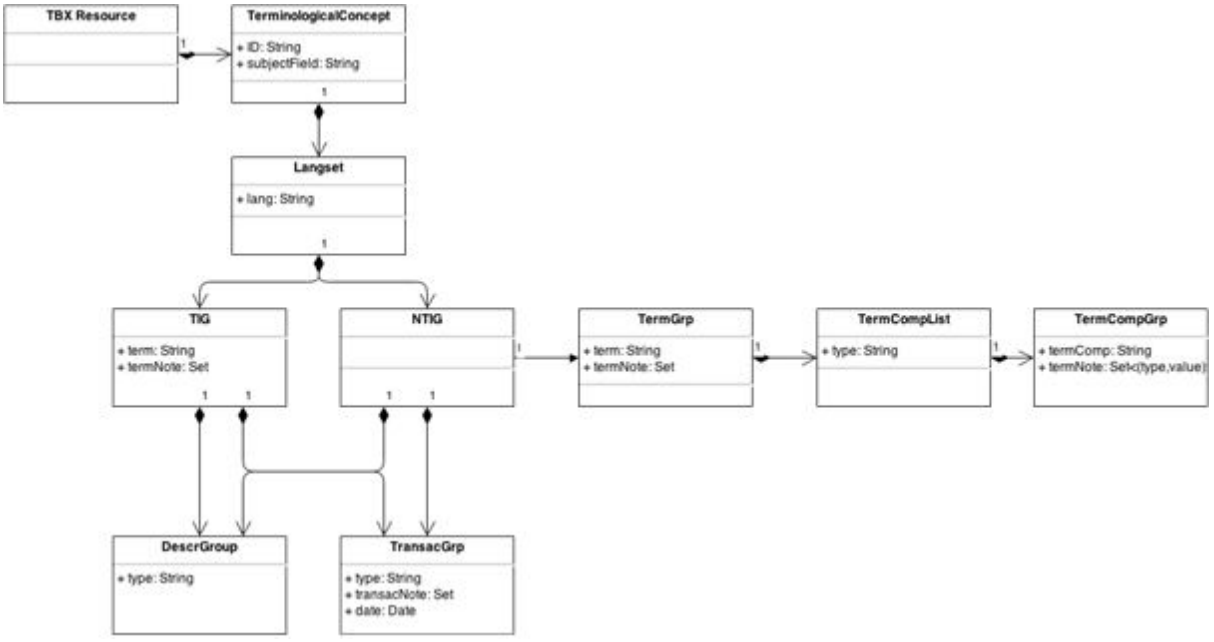


Figure 11: TBX Data Model diagram

The main elements in the TBX Data Model are:

- **TBX Resource:** A TBX resource essentially represents a collection of terminological concepts (**Terminological Concept**), which are represented as XML elements of type *termEntry* and have a unique ID. In the above XML snippet, there is one terminological concept with ID 2151845. Each terminological concept is described by a set of properties, such as a *subject field* they belong to.
- **Terminological Concept:** represents a language-independent concept. Each terminological concept is associated to a *LangSet*, which can be seen as a set of language-specific **Terms** that express the **Terminological Concept** in question.
- **Langset:** A langset is a language-specific container for all the terms that lexicalize a Terminological Concept in a given language. The **Langset** contains simple terms, for which no decompositions is provided (TIG), as well as complex terms for which the decomposition information is provided (NTIG).
- **TIG:** represents a language-specific term for which no decomposition information is provided.
- **NTIG:** represents a language-specific term for which decomposition information is provided.
- **TermGrp:** contains information about a language-specific term including ist morphosyntactic properties; there is one TermGrp for each TIG and NTIG

- **TermCompList:** represents the decomposition of a term
- **TermCompGrp:** represents one component of a term and its morphosyntactic properties
- **DescrGrp:** describes the properties of a particular term, in particular different surface forms or describes contexts that document the usage of the term
- **TransGrp/Transaction:** contains information about a transaction that lead to the creation or modification of a term.

The main data elements described above have been mapped into RDF using a set of vocabularies in order to reuse already existing classes and properties (see Table 2).

Table 2 - Classes and Properties reused from other vocabularies

Vocabulary Name	Abbr.	URL	Reused Elements
Resource Description Framework	rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns`	**Properties:** `type, _1, _2, _3`
Resource Description Framework Schema	rdfs	`http://www.w3.org/2000/01/rdf-schema`	
Dublin Code	dc	`http://purl.org/dc/terms`	**Properties:** `source, creator`
SKOS	skos	`http://www.w3.org/2004/02/skos/core`	**Classes:** `Concept`
Provenance	prov	`http://www.w3.org/ns/prov`	**Classes:** `Activity, Entity` **Properties:** `endedAtTime wasAssociatedWith wasGeneratedBy`
Lexicon Model for Ontologies – Core Module	ontolex	`http://www.w3.org/ns/ontolex`	**Classes:** `Lexicon, LexicalEntry, LexicalSense` **Properties:** `language, canonicalForm lexicalizedBy, entry, definition, writtenRep, otherForm, sense`

Lexicon Model for Ontologies – Decomposition Module	decomp	`http://www.w3.org/ns/lemon/decomp`	**Classes:** `Component` **Properties:** `constituent` `identifies`
Vocabulary of Interlinked Datasets	void	`http://www.w3.org/TR/void/`	**Classes:** `Dataset`

TBX Data Model elements have been mapped to the above classes and properties, as it follows:

- **TBX Resource:** is not explicitly represented, the whole dataset represents the TBX resources. A TBX resources is thus represented as a **`void:Dataset`**. Provenance information is attached, specifying that the data has been converted by the LIDER converter.
- **Terminological Concept:** is represented as a **`skos:Concept`**
- **Langset:** A langset is not represented as such in the data. Instead, one **`ontolex:Lexicon`** is created for each language for which a Langset is defined. The collection of all the terms for a given language will belong to the corresponding language-specific **`ontolex:Lexicon`**
- **TIG/NTIG:** are represented as **`ontolex:LexicalEntry`**, no distinction is made between terms with decomposition and terms without decomposition; if no decompositions information is available, this is simply omitted. In that sense the representation is monotonic as the decomposition information can be added later
- **TermGrp:** the information about the morphosyntactic properties of a term is attached to the corresponding **`ontolex:LexicalEntry`**. The string enclosed in *<term> </term>* is assumed to be the **`ontolex:canonicalForm`** of the lexical entry in question.
- **TermCompList:** the decomposition of a term is represented using the **`ontolex:decomp`** vocabulary, creating a **`decomp:Component`** and **`ontolex:LexicalEntry`** for each component.
- **TermCompGrp:** the morphosyntactic properties of a component are attached to the corresponding lexical entry that is identified (through **`decomp:identifies`**) with the component in question)
- **DescrGrp:** descriptions of the term or context are mapped to appropriate properties of the lexical entry or the context
- **TransGrp/Transaction:** a transaction that creates or modifies the term is mapped to a **`tbx:Transaction`** (a subclass of **`prov:Activity`**). Provenance metadata is attached to this entity. The **`prov:Activity`** related to the responsible person or agent through **`prov:wasAssociatedWith`**; the relation to the responsible Agent is encoded via **`prov:wasGeneratedBy`**.