



**HAL**  
open science

# A Semantic and Smart Framework for Handling Multilingual Linguistic Knowledge: A Framework and Case Implementation

Mariem Neji, Fatma Ghorbel, Bilel Gargouri, Nada Mimouni, Elisabeth Metais

► **To cite this version:**

Mariem Neji, Fatma Ghorbel, Bilel Gargouri, Nada Mimouni, Elisabeth Metais. A Semantic and Smart Framework for Handling Multilingual Linguistic Knowledge: A Framework and Case Implementation. International Journal of Information Technology and Web Engineering, 2022, 17 (1), 10.4018/IJITWE.314571 . hal-03885570

**HAL Id: hal-03885570**

**<https://hal.science/hal-03885570v1>**

Submitted on 6 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# A Semantic and Smart Framework for Handling Multilingual Linguistic Knowledge: A Framework and Case Implementation


Mariem Neji, MIRACL, Tunisia\*

Fatma Ghorbel, MIRACL, Tunisia

Bilel Gargouri, MIRACL, Tunisia

Nada Mimouni, ISID, CEDRIC, France

Elisabeth Métais, CNAM, France

 <https://orcid.org/0000-0002-2016-4198>

## ABSTRACT

The authors propose a semantic and smart assistance framework for handling linguistic knowledge, called LingFramework. It targets both expert and novice users. It aims to assist the user in understanding the different aspects of the linguistic domain and ease the process of proposing lingware applications. LingFramework is based on a multilingual linguistic domain ontology called LingOnto. It allows (1) representing linguistic data, linguistic processing functionalities and linguistic processing features, and (2) reasoning, via a SWRL-based reasoning engine, about the linguistic knowledge. Currently, it covers English, French, and Arabic languages. To facilitate the interaction with LingOnto, an ontology visualization tool called LingGraph is proposed. It offers an easy-to-use interface for users not familiar with ontologies. It provides a SPARQL pattern-based approach to allow a smart search interaction functionality. LingFramework is applied to assist the user in identifying valid linguistic processing pipelines related to lingware applications. The evaluation results are promising.

## KEYWORDS

Linguistic Domain Ontology, Multilingualism, Natural Language Processing, Smart Framework, User Friendly Visualization

## 1. INTRODUCTION

Natural Language Processing (NLP) has received a lot of attention as a way to enhance the development of computer applications and techniques capable to automatically processing data relating to one or more languages (Zhou et al., 2020). Assimilating the linguistic domain is a fundamental issue; especially that it is involved in many other research domains such as e-health, e-business, education and antiterrorism. For instance, NLP is used in the e-health domain by proposing human-to-machine natural language instruction such as, robot-assisted surgery guided by human instruction (Costea et al., 2020). However, the linguistic domain is huge and complex and it presents considerable differences between languages (Schalley, 2019). These issues become worse when handling lingware applications (Baklouti et al., 2010).

DOI: 10.4018/IJITWE.314571

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Representing the linguistic knowledge in a common model could help the user to understand the meaning, scope and use of related techniques and algorithms. This is particularly useful for novice users, but can also provide new perspectives for expert ones.

Various linguistic registries and glossaries have been proposed. Unfortunately, such efforts provide a poor and an imprecise semantic description which are not sufficient for most lingware applications. Besides, they do not support multilingualism. Ontologies were proven to be more useful as they provide more precise and semantically richer results (Jarrar, 2021). However, most of the proposed ontologies represent only the linguistic data (e.g. word and Part Of Speech (POS)) and neglect the linguistic processing functionalities (e.g. segmentation and POS tagging) and the linguistic processing features (e.g. processing level and analysis type). Moreover, they do not offer a reasoning engine that assists the user in understanding the linguistic knowledge and developing lingware applications. Besides, they are hard to be used by users less or not familiar with ontologies as they do not offer an ontology visualization tool to facilitate the interaction with it. Finally, most of these ontologies do not support multilingualism.

In this paper, the authors propose LingFramework, a semantic and smart assistance framework for handling multilingual linguistic knowledge. It targets not only expert users, but also novice ones. It assists users in understanding the different aspects of the linguistic domain and ease the process of developing NLP applications. LingFramework is based on a multilingual linguistic domain ontology called LingOnto. This ontology allows representing linguistic data, linguistic processing functionalities and linguistic processing features. Moreover, LingOnto enables reasoning, via a SWRL based reasoning engine, about the aforementioned knowledge in order to guide the user to select valid NLP pipelines. For example, if the user is developing an annotation tool, he will be guided through each processing functionality choice, where only functionalities that are valid for the annotation task in the processing pipeline are made available for selection. LingOnto covers the French, English and Arabic languages. LingOnto is designed to be used by users, who are not necessary ontology experts. To overcome this issue, the authors propose a user friendly ontology visualization tool called LingGraph. It offers an understandable visualization of LingOnto to both ontology and non-ontology expert users. LingGraph is based on a smart search functionality which relies on a SPARQL pattern-based approach. It extracts and visualizes an ontological view from LingOnto related to only components corresponding to the user's needs.

The authors applied LingFramework to assist users in identifying valid NLP pipelines related to NLP applications. Finally, they evaluate (i) its performance in identifying valid NLP pipelines and (ii) the usability of its user interfaces.

The current paper is organized as follows. Section 2 presents some related works. Section 3 details our semantic and smart framework for handling linguistic knowledge. Section 4 presents our multilingual linguistic domain ontology. Section 5 presents our user friendly ontology visualization tool. Section 6 details a brief example of a typical use of LingFramework. The evaluations of the performance and usability of our framework will be presented in Section 7. Finally, Section 8 draws conclusions and future research directions.

## 2. RELATED WORK

The present work is closely related to the following research areas: (i) linguistic knowledge representation and (ii) ontology visualization.

### 2.1. Linguistic Knowledge Representation

Various approaches focusing on linguistic knowledge representation are proposed. The authors distinguish two main categories: linguistic bases-based approaches and ontologies-based approaches.

### *2.1.1. Linguistic Bases-Based Approaches*

The SIL glossary of linguistic terms (Loos, 2004) represents information based on glossaries and bibliographies proposed to support the linguistic research. This glossary supports only French and English linguistic terms. Moreover, it gives only the equivalent(s) of a linguistic term in the other language (i.e., it gives English glosses for French linguistic terms and French glosses for English linguistic terms). Furthermore, the relations between linguistic terms are unspecified or too general to derive the meaning of a linguistic concept within the NLP domain (Chiarcos and Hellmann, 2017).

The ISOcat data category registry (Ide and Romary, 2004) defines only linguistic data at several levels, such as syntactic, morphosyntactic, terminological and lexical. However, navigating through it is a tedious task since it provides a wide range of different “views” and “groups” that specifies linguistic data in a specific language data model. In this regard, the ISOcat data category registry has no underlying data model that represents linguistic data in an interrelating holistic structure.

In attempts to define linguistic terms in a stricter manner, the CLARIN Concept registry (Schuurman et al., 2016) takes over the work of the ISOcat data category registry. However, this latter still provides very limited structural and relational information (Schuurman et al., 2016).

The authors note that in all the above-mentioned linguistic registries, the structure of the data models representing the linguistic data entries in alphabetical order (e.g., the SIL glossary) or according to linguistic views (e.g., the ISOcat) is not sufficient for ensuring comprehensive knowledge about a linguistic data in the NLP domain. Moreover, they focus only on representing the linguistic data aspect and neglect the processing one. Finally, they define a flat semantic structure providing very unspecific relations between concepts such as “is\_a” or “has\_kinds” (Chiarcos and Hellmann, 2017).

### *2.1.2. Ontologies-Based Approaches*

In (Fellbaum, 1998), the authors propose WordNet, which contains an extensive taxonomic and mereological structure that could be regarded as a kind of proto-ontology. However, its object properties are not used in a consistent way as they present redundancy (Gangemi et al., 2002). Moreover, it provides a poor classification of the types of numbers (Jarrar, 2021).

In (Farrar and Langendoen, 2010), the authors propose the General Ontology for Linguistic Description (GOLD). It provides a taxonomy of nearly 600 concepts, 76 object properties and 7 data properties. However, most of the object properties interrelate only two concepts, which leaves the majority of the concepts unrelated. Moreover, this ontology does not aim to capture the semantics of terms. It mainly classifies morphological notations, such as expressions, grammar, and meta-concepts (Jarrar, 2021). The development of this ontology was stopped in 2010.

In (Chiarcos and Sukhareva, 2015), the authors propose the Ontologies of Linguistic Annotation (OLiA), which is based on the ISOcat data category registry and the GOLD ontology. It takes a focus only on modeling annotation schemes and their linking with reference categories. Conceptually, the OLiA ontology is closely related to the OntoTag ontologies (de Cea et al., 2004). One important difference is that the OntoTag<sup>1</sup> ontologies are considering only the languages of the Iberian peninsula (in particular Spanish).

In (Jarrar, 2021), the author proposes a linguistic ontology for the Arabic language, which is a formal representation of the concepts that the Arabic terms convey. This ontology is considered as an “Arabic WordNet” as it uses the same structure. It consists currently of about 1,000 well investigated concepts in addition to 11,000 concepts that are partially validated. However, this ontology does not support multilingualism as it considers only the Arabic language.

In (Klimek et al, 2021), the authors propose the Multilingual Morpheme Ontology called MMoOn Core. The aim of this ontology is to serve as a shared semantic model for linguists and NLP researchers alike to enable the creation, conversion, exchange, reuse and enrichment of morphological language data across different data-dependent language sciences. However, this ontology focuses only on representing the linguistic data aspect and neglects the processing one.

In (Laatar et al, 2022), the authors propose an historical ontology for the Arabic language. The aim of this ontology is to capture the evolution of Arabic words and their diverse meaning over centuries. However, this ontology does not support multilingualism as it considers only the Arabic language.

The authors note that all the above-mentioned ontologies focus only on representing linguistic data aspect and neglect the processing one. Furthermore, they do not propose a reasoning mechanism. Besides, they are hard to be used by users less or not familiar with ontologies as they do not offer an ontology visualization tool to facilitate the interaction with it. Finally, most of these ontologies do not support multilingualism.

## 2.2. Ontology Visualization

In the literature, various ontology visualization tools are proposed. However, most of them are designed to be used only by ontology expert users and they overlook the importance of the usability and understandability requirements. According to (Lohmann et al., 2016), the generated visualizations “are hard to read for casual users”. For instance, GrOWL and SOVA<sup>2</sup> are intended to offer an understandable visualization by defining notations using different symbols, colors, and node shapes for each ontology key-element. However, the proposed notations contain many abbreviations and symbols from the Description Logic. As a consequence, the generated visualizations are not suitable for non-ontology expert users. OWLViz<sup>3</sup>, OntoTrack (Liebig and Noppens, 2005), KC-Viz, OntoViz and MLGrafViz (Florence et al, 2021) show only specific element(s) of the ontology. For instance, the OWLViz and KC-Viz visualize only the class hierarchy of ontology and OntoViz shows only inheritance relationships between the graph nodes. This is different with TGViz Tab (Alani, 2003) and NavigOWL (Hussain et al., 2014) and BioOntoVis (Achich et al, 2018) which provide visualizations representing all the key elements of the ontology. However, these tools do not make a clear visual distinction between the different ontology key-elements. For instance, they use a plain node-link diagram where all the links and nodes look the same except for their color. This issue has a bad impact on the understandability of the generated visualization.

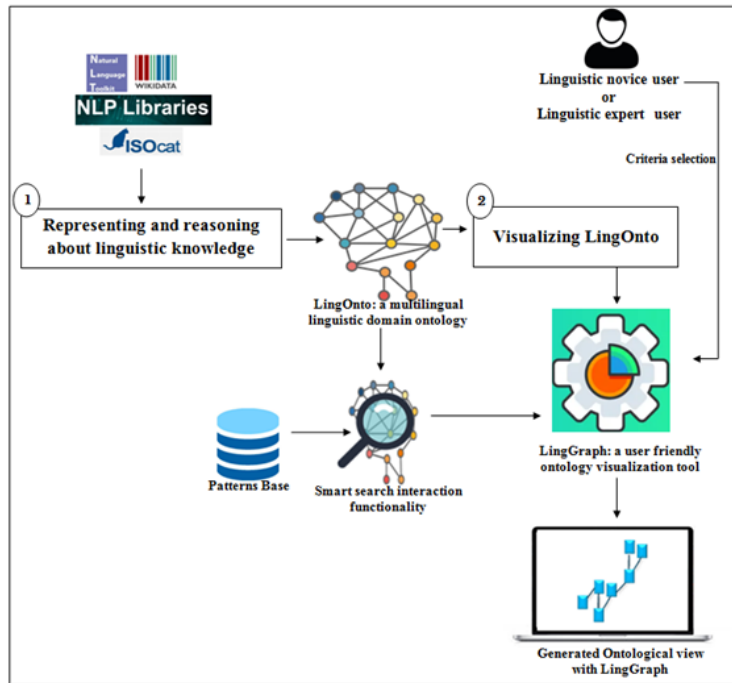
Only very few visualization tools are designed to be used by non-ontology experts such as OWLeasyViz (Catenazzi et al., 2009), Protégé VOWL (Lohmann et al., 2016) and WebVOWL (Lohmann et al., 2016). However, these efforts are either not available for downloading, such as OWLeasyViz or using some Semantic Web words such as WebVOWL and ProtégéVOWL.

Most of these tools offer a basic keyword-based search interaction technique. It is based on a simple matching between ontology’s elements and the keyword that the user is looking for. However, they do not offer advanced search by extracting a combination of components taking into account the user’s need.

## 3. OVERVIEW OF LINGFRAMEWORK

LingFramework is a smart framework for handling multilingual linguistic knowledge. It targets not only expert users, but also the novice ones. The aim is to assist the user in understanding the different aspects of the NLP domain and ease the process of proposing lingware applications. LingFramework is based on a multilingual linguistic domain ontology called LingOnto, that aims to represent and reason about linguistic knowledge. To facilitate the interaction with this ontology, the proposed framework offers a user-friendly ontology visualization tool. This latter offers an understandable visualization to users less/not familiar with ontologies. Furthermore, it provides a set of interaction techniques such as zooming, graph browsing and filtering nodes. Compared to available ontology visualization tools that offer static keyword-based search interaction technique, LingGraph offers a “smart search” interaction functionality. It relies on a SPARQL pattern-based approach that aims to extract and visualize an excerpt ontological view, from LingOnto. This latter contains only components corresponding to the user’s needs. LingFramework is synthesized in Figure 1 and is more detailed in the following sections.

Figure 1. Overview of LingFramework



## 4. LINGONTO ONTOLOGY: REPRESENTING AND REASONING ABOUT LINGUISTIC KNOWLEDGE

The authors propose a multilingual linguistic domain ontology called LingOnto. It is freely available online <sup>4</sup>. The current version of LingOnto covers the English, French and Arabic languages. Compared to related work, it does not only handle linguistic data, but also linguistic processing functionalities and linguistic processing features. Besides, it allows via a reasoning engine, inferring new linguistic knowledge from those initially entered and assisting in the process of proposing lingware applications.

### 4.1. Representing Linguistic Knowledge

The authors are based on the design principles defined by (Gruber, 1995), which are objective criteria for proposing and evaluating ontology designs, such as clarity, coherence, minimal encoding bias and minimal ontological commitments. Following these principles, the authors define the top-level concepts of our ontology which are linguistic data, linguistic processing functionalities and linguistic processing features. These latter will be more discussed in the following sections.

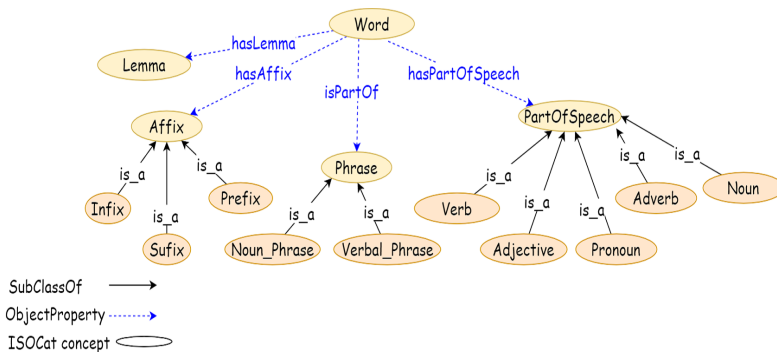
#### 4.1.1. Linguistic Data Classification

Referring to the ISOcat standard, the authors identify a set of linguistic data concepts. They choose this registry for the following advantages:

- It covers more terms of linguistic data categories compared to other resources.
- It defines linguistic data categories at several levels.
- It supports multilinguism.

For each extracted linguistic data concept, the authors identify the concepts that are related to it as well as the names of the associated relations. Figure 2 shows an excerpt of LingOnto, illustrating the classification of some Arabic linguistic data. Indeed, in contrast to the English sentences which are fundamentally in the (subject–verb) order, the Arabic ones can be nominal (subject–verb), or verbal (verb–subject) with a free order. Thus, the authors define an “is\_a” object property relating the (“Phrase”) class and (“Noun\_Phrase”) and (“Verbal\_Phrase”) classes. Furthermore, in French and English languages, the affix is classified into prefixes, suffixes, infixes, circumfixes, and superfixes. However, in the Arabic language, the affix is classified only into prefixes, suffixes and infixes. Consequently, the authors define an “is\_a” object property between the (“Prefix”), (“Suffix”) and (“Infix”) classes and (“Affix”) class. Moreover, Arabic differs phonetically, morphologically, syntactically and semantically from English and French languages. For instance, Arabic has a rich and complex inflectional morphology involving: gender, number, person, aspect, mood, case, state and voice, cliticization of a number of pronouns and particles (e.g., conjunctions, prepositions and definite article). Syntactically, the Arabic sentences are too long with a complex syntax compared to the English and French languages (e.g., a single verbal sentence can consist of more than 50 words).

Figure 2. The classification of some Arabic linguistic data



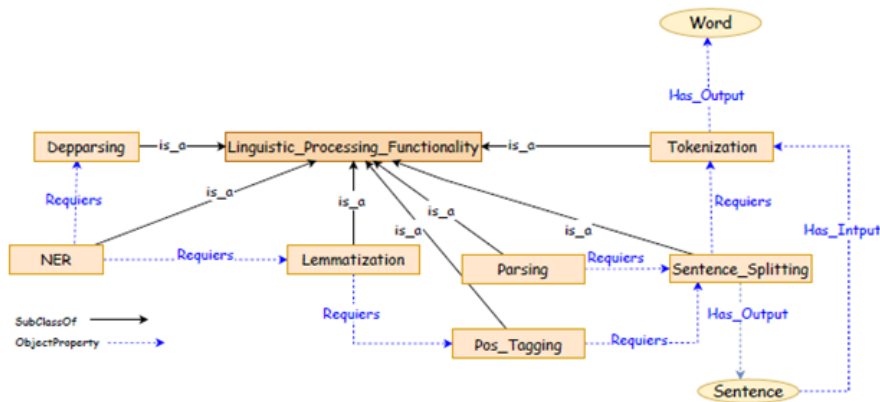
#### 4.1.2. Linguistic Processing Functionalities Classification

Referring to well-known NLP toolkits such as Apache OpenNLP (Mohan and Samuel, 2016), StanfordCoreNLP (Manning et al., 2014), FreeLing (Atserias et al., 2006) and LingPipe (Konchady, 2008) and two language processing platforms which are Language Grid (Ishida, 2006) and Gate (Fairen Jimenez et al., 2011), the authors identify a set of linguistic processors such as “POS Tagger”, “Lemmatizer”, “Morphological Analyzer” and “Chunker”. Some of these linguistic processors often implements one or two linguistic processing functionalities. For instance, a “Morphological Analyzer” processor for French and English languages usually implements “Paragraph splitting”, “Sentence splitting”, “Tokenization”, “POS tagging” and “Lemmatization” functionalities. Nevertheless, a “Morphological Analyzer” processor for Arabic language and especially for analyzing diacritic Arabic texts implements “Paragraph splitting”, “Sentence splitting”, “Tokenization”, “POS tagging”, “Diacritization” and “Lemmatization” functionalities. Therefore, the automatic diacritization is an essential processing functionality for many Arabic lingware applications. Moreover, Arabic sentence components can be swapped without affecting the structure or meaning. For this reason, it leads to a more syntactic and semantic ambiguity in contrast to the English and French languages.

According to (Hayashi and Narawa, 2012), it exists an hierarchical inter-dependencies between the linguistic processing functionalities. Indeed, a linguistic processing functionality used to perform a given analysis at one level may require, as input, the results of others analysis related to a lower

level. For instance, to annotate a French text, this latter must be tokenized, the sentences should be clearly separated from each other and their morphological properties have to be analyzed before starting the parsing functionality. Consequently, the authors identify the object property “Requires”. As shown in Figure 3, the (“Tokenization”) class is in relation with the (“Sentence-Splitting”) class through the object property “Requires”. Moreover, each functionality uses various linguistic data as inputs and others as outputs. Hence, the authors propose the object properties “Has-Input” and “Has-Output”. For instance, as shown in Figure 3, the (“Tokenization”) class is in relation with the (“Sentence”) class through “HasInput” object property. It is also in relation with the (“Word”) class through “Has-Output” object property.

Figure 3. The classification of some Arabic linguistic processing functionalities



#### 4.1.3. Linguistic Features Classification

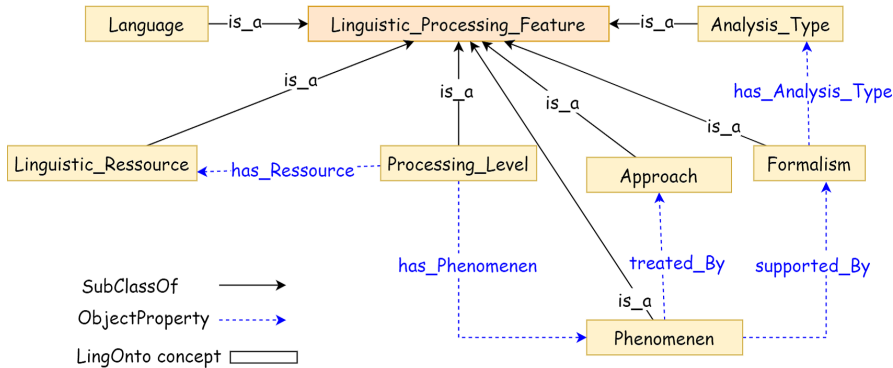
The linguistic processing functionalities are characterized by several linguistic features. LingOnto models these features to ease the process of proposing lingware applications as they identify the incoherence between linguistic processing functionalities.

The English, French and Arabic languages are based on the same linguistic processing features. Indeed, according to (Haddar and Hamadou, 2009), a comparative study of English, French and Arabic sentences shows that it is possible, from the linguistic viewpoint, to adopt the same typology of ellipses (i.e., Gapping, Right-node Raising, Coordination Reduction) for the Arabic language as the one proposed for the English and French languages.

Figure 4 shows the proposed classification of the linguistic processing features. Each processing level is characterized by its related phenomena. Hence, the authors define the object property “has Phenomenon” between (“ProcessingLevel”) and (“Phenomena”) classes. Moreover, each phenomenon has its sub phenomena. For example, the ellipsis phenomenon can be a nominal ellipsis or an ellipsis of the whole sentences. For this reason, the authors define the “refined into” reflexive object property. The linguistic phenomenon has also the relations “supportedBy” and “treatedBy”, respectively, with the (“Formalism”) and (“Approach”) classes. Each formalism has an analysis type to solve any linguistic phenomenon. For example, the sentence “*Jean dropped the plate. It shattered loudly.*” illustrates the Anaphora phenomenon. In this sentence, the pronoun “it” is an anaphor and it points to the left toward its antecedent “the plate”. Finally, each processing level uses a linguistic resource related to a phenomenon. Hence, the authors define the object property “has Resource” relating to the (“Processing Level”) and (“Linguistic Resource”) classes.



Figure 4. The classification of some linguistic processing features



## 4.2. Reasoning About Linguistic Knowledge

LingOnto proposes a set of SWRL rules to reason about linguistic knowledge, infer new data from those initially entered and assist the user in understanding the linguistic NLP domain. The authors categorize the proposed SWRL rules into two categories:

- **SWRL rules for lingware applications development assistant:** LingOnto proposes a set of SWRL rules that assist the user in selecting compatible linguistic processing functionalities in order to identify valid NLP pipelines:
  - **Rule R1:** identifies if a processing functionality “x” requires a processing functionality “y” and a processing functionality “z” requires a processing functionality “x”, then add a requires relation between the processing functionalities “z” and “y”.
  - **Rule R2:** identifies if a processing functionality “x” has as input a linguistic data “i” and a processing functionality “y” has as output a linguistic data “i”, then add a requires relation between the processing functionalities “x” and “y”.
  - **Rule R3:** identifies if a processing functionality “x” requires a processing functionality “y” and the processing functionality “x” uses a linguistic resource “j” and the processing functionalities “x” and “y” belong to the same linguistic processing level then the processing functionality “y” uses the linguistic resource “j”.
- **SWRL rules for NLP domain understanding assistant:** LingOnto proposes a set of SWRL rules to assist the user in understanding the meaning of different linguistic knowledge:
  - **Rule R 4:** identifies if a phrase “x” has a main part a verb “y”, then the phrase “x” is a verbal phrase.
  - **Rule R 5:** identifies if an affix “y” surrounds a stem “y”, then the stem “y” is a circumfix.
  - **Rule R 6:** identifies if a word “x” has a gender neuter, then the word “x” is in English.

## 5. LINGGRAPH: ONTOLOGY VISUALISATION TOOL OF LINGONTO

The LingOnto domain ontology is designed to be used by users, who are not necessary ontology experts. To overcome this issue, the authors propose a user friendly ontology visualization tool called LingGraph. It is freely available online<sup>5</sup>. It offers an understandable visualization to both ontology and non-ontology expert users. Moreover, it is based on a smart search functionality. This latter relies on a SPARQL pattern-based approach. The aim is to extract and visualize an excerpt ontological view, from LingOnto, that contains only components (i.e., concepts and object properties) corresponding to

a specific user need. This latter is materialized by a set of predefined search criteria  $C = (C_1, \dots, C_n)$ . Currently, the authors identified three search criteria which are: (“Abstraction Level”), (“Processing Level”) and (“Language”). For each criterion  $C_i$  ( $i \in [1, n]$ ), a set of preferences  $CP = (C_{p_{i/1}}, \dots, C_{p_{i/m}})$  is associated. For example, the preferences associated with the criterion Processing Level are: (“lexical level”), (“morphological level”), (“semantic level”) and (“syntactic level”). The user can select more than one preference of each criterion.

The authors asked some users (expert and novice users) to fill a pre-questionnaire about what they need to know about linguistic knowledge. The authors notice that user’s needs seem very regular as all of them wonder what the abstraction level (e.g., linguistic data and/or processing functionalities and features) of such a processing level(s) or/and such language(s). This is a basic observation that naturally leads us to the idea of patterns. The authors propose a set of SPARQL patterns  $P = (P_1, \dots, P_k)$  that is designed, manually, by a domain expert. The following section gives a definition of a pattern and details some examples.

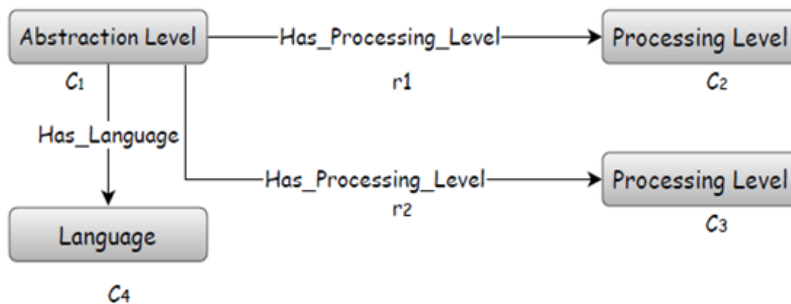
### 5.1. Pattern Definition

A pattern  $P$  is a couple  $(G, Q)$  such as:

- $G$  is a connected RDF graph  $G$  which describes the general structure of the pattern and represents a family of queries. Such a graph only contains triples according to the structure presented in Figure 6.
- $Q$  represents the qualifying elements that characterize the pattern and will be taken into account during the mapping of the user query and the considered pattern. A qualifying element can either be a vertex (representing a class or a datatype) or an edge (representing an object property or a datatype property) of  $G$ .

Figure 5 display the pattern covering the need: [ $C_1 =$  “Abstraction Level”,  $CP_{1/1} =$  “Processing Functionalities”], [ $C_2 =$  “Processing Level”,  $CP_{1/2} =$  “Lexical Level”,  $CP_{2/2} =$  “Morphological Level”], [ $C_3 =$  “Language”,  $CP_{1/3} =$  “Arabic”]. In this pattern, the vertex  $C_1$  and  $c_2$  and the arc  $r_1$  are called qualifying elements. Each vertex defines a selected criterion  $C_i$  (i.e., vertex  $C_1$  defines the selected criterion (“Abstraction level”) and the vertex  $C_2$  defines the selected criterion (“Processing Level”). Each vertex must be replaced by another resource, in order to turn the pattern into a query. This means that, to obtain the query graph corresponding to the user need, each vertex must be substituted by the selected preferences of the concerned selected search criterion. Each preference  $CP_j/i$  ( $j \in [1, n]$ ) has a corresponding concept in LingOnto having the same name. This process is called an instantiation.

Figure 5. An Example of a pattern



## 5.2. Pattern Instantiation

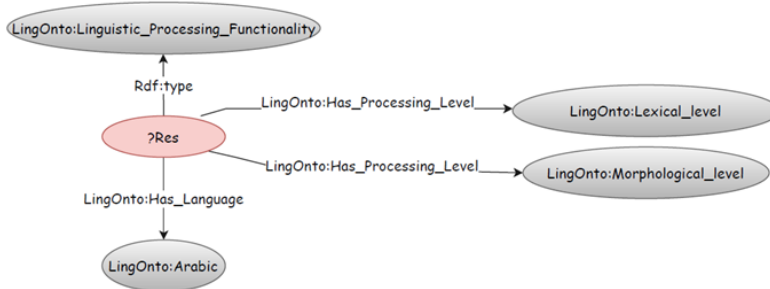
In this section, we explain the instantiation of a qualifying element of a pattern. In other words, we will see how the query graph is transformed when one of its qualifying elements is brought closer to an element of the user's need.

For all  $q$  qualifying elements of  $p(G,Q)$  and  $\alpha$  extracted from the user request (which can be either a class, an instance, or a property), we denote by  $I(p,q,\alpha) = (G_0, Q_0)$  the pattern obtained after the instantiation of  $q$  by the resource  $\alpha$  in the pattern  $p$ . This instantiation is only possible if  $q$  and  $\alpha$  are compatible:

- $q$  is a class and  $\alpha$  an instance of  $q$ . Then the instantiation of the qualifying concept consists in replacing the URI of the class by the URI of the instance.
- $q$  is a datatype and  $\alpha$  a value corresponding to the type  $q$ . Then the instantiation of the qualifying concept consists in replacing the URI of the class by the value  $\alpha$ .
- $q$  is a property and  $\alpha$  the same property or one of its sub-properties. Then the instantiation of the qualifying edge consists in replacing the URI of the edge by the URI of the property  $\alpha$ .

The instantiation of the pattern shown in Figure 5 leads, after substitution of each qualifying element by the selected preferences, to the query graph shown in Figure 6.

Figure 6. Query graph resulting from the instantiation of the pattern of Figure 5



## 5.3. Generation of the SPARQL Query

A question mark in front of an element means that this element is one of the objects of the query. Therefore, the authors find the qualifying vertices associated with these query elements in the SELECT clause of our SPARQL query. For each query element preceded by a question mark:

If the qualifying vertex in question refers to a class or a data type, it has already been replaced by a variable in the previous step, so the authors add this same variable in the SELECT clause. Otherwise (the qualifying vertex refers to a relation) it is a request for specialization or generalization of a relation. In this case, the qualifying vertex is replaced in the query graph by a variable, explicitly declared as a sub-property or super-property of the relation referenced by two triplets made alternative in SPARQL with UNION, this variable is also added in the SELECT clause.

The authors have thus identified all the elements of the graph on which the query is based and obtained the definitive query graph which will form the content of the WHERE clause of our query. Figure 7 shows the generated SPARQL query corresponding to the query graph in Figure 6.

Figure 7. The generated SPARQL query associated with the request graph shown in Figure 6

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?Linguistic_Processing_Functionality
WHERE {
    ?Linguistic_Processing_Functionality rdfs:has_processing_level ?morphological_level.
    ?Linguistic_Processing_Functionality rdfs:has_processing_level ?lexical_level.
    ?Linguistic_Processing_Functionality rdfs:has_Language ?Arabic
}
```

## 6. EXPERIMENTATION

In this section, the authors present a demonstrative example of LingFramework in identifying valid NLP pipelines associated with a lingware application. Then the authors evaluate the usefulness and usability of the proposed framework.

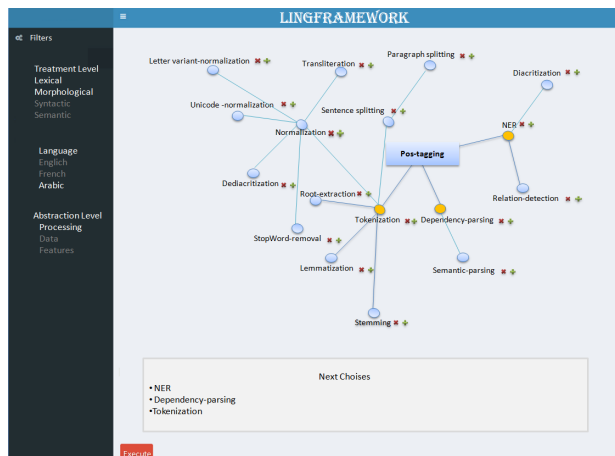
### 6.1. A Demonstrative Scenario

Lingware applications typically consist of many individual components that need to be used together in order to solve real-world problems (Ziad et al., 2018). However, the combination of multiple components in a particular order into a processing pipeline is a tedious task which can be a barrier for domain experts and especially for novice ones. LingFramework is applied to assist the user in identifying valid pipelines associated with a Morphological Analyzer for Arabic texts.

As shown in Figure 8, the user starts by selecting the preferences “Lexical level” and “Morphological level” as a Processing Level, “Arabic” as a Language and “Linguistic processing” as an Abstraction level. Consequently, based on the smart search interaction functionality, an excerpt ontological view corresponding to the expressed need is generated.

Then, the user starts the process of identifying an NLP pipeline solving the target lingware application. Consequently, LingFramework offers, under “Next choices”, a set of possible processing functionalities which can be added after each selected functionality. This list is generated based on the predefined SWRL rules. For instance, Figure 8 shows that after a “Pos-tagging” functionality, only “NER”, “Dependency-parsing” or “Tokenization” functionalities may be added. These latter can be added to the pipeline by double-clicking on them.

Figure 8. NLP pipeline construction screenshot



If the user selects a processing functionality out of the list under “Next Choices”, LingFramework displays an error message “Incompatible Functionalities” and indicates using the red color an alternative valid pipeline. As shown in Figure 9, the “Diacritization” functionality can be added to the pipeline only after “Pos-tagging” and “NER” functionalities.

The final NLP pipeline is shown in Figure 10.

Figure 9. Alternative NLP pipeline proposition screenshot

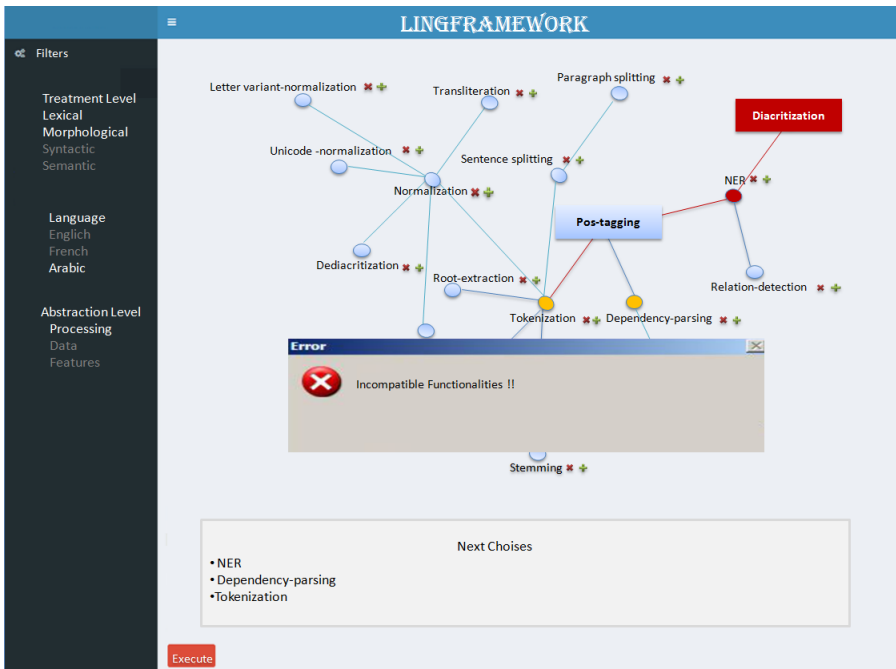
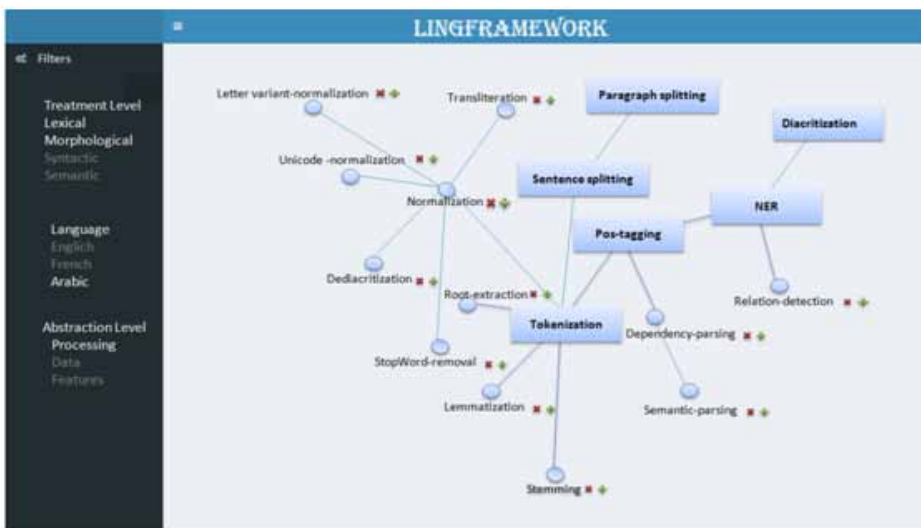


Figure 10. The final NLP pipeline



## 6.2. Evaluation of LingFramework

The authors evaluate the usefulness and the usability of LingFramework. They have two main aims. The first one is to evaluate its performance in identifying valid NLP pipelines associated with lingware applications. The second aim is to evaluate the usability of its user interface (i.e., usage easiness and non requirement of expertise).

### 6.2.1. Evaluation of the Performance

The evaluation of the performance of LingFramework in identifying valid NLP pipelines consists of three steps:

- **Step 1:** the authors prepared 63 NLP tasks, which had to be solved using LingFramework. They classify these tasks into (1) Low level processing tasks and (2) High level processing tasks. Further, they classify NLP tasks in each group according to the language. Then, they classify applications in each group according to the language (i.e., French, English and Arabic). Table 1 shows some examples of this classification.

Table 1. Examples of proposed lingware applications

Language	Low Level lingware application	High Level lingware application
French	A Co-reference resolver, A chunker	A text summary generator A sentiment analyses resolver
English	A text annotator, An inflected words reducer	An inference resolver Relevant terms extractor
Arabic	An inflectional endings remover A morphological analyzer	A question answer A text summary generator

- **Step 2:** the authors recruit three linguistic experts. The first one is a member of the Arabic Natural Language Processing Research Group (ANLP-RG) of MIRACL laboratory (Tunisia, Sfax). The second is a member of the CEDRIC laboratory (France, Paris). The last expert is a member of the Formal linguistics laboratory (France, Paris). The authors ask each expert to provide, manually, all the possible pipeline(s) which may solve each lingware application according to their native language (i.e., French, English and Arabic).
- **Step 3:** the authors identify, using LingFramework, all the possible NLP pipeline(s) corresponding to each lingware application identified in Step1. Then, the experts provide their feedback according to each generated pipeline (“Valid or Not valid” pipeline). The experts may also provide a textual explanation.

The authors use the precision and recall metrics (Su, 1994) to evaluate the performance of the LingFramework. The recall measures the proportion of valid pipelines which have been identified using LingFramework among identified pipelines by the domain expert. The precision measures the proportion of valid pipelines identified using LingFramework within the total number of identified pipelines. The authors evaluate the performance of LingFramework in identifying valid pipelines associated with the low and high level proposed applications treating one language as shown in Figure 11 and Figure 12.

The precision and recall metrics indicate that LingFramework is efficient in identifying valid NLP pipelines for high and low processing levels. Indeed, as shown in Figure 11 the overall means of the precision associated with the English and French languages (86.3% and 92.3%) are almost

Figure 11. Recall and Precision performances for low-level lingware applications

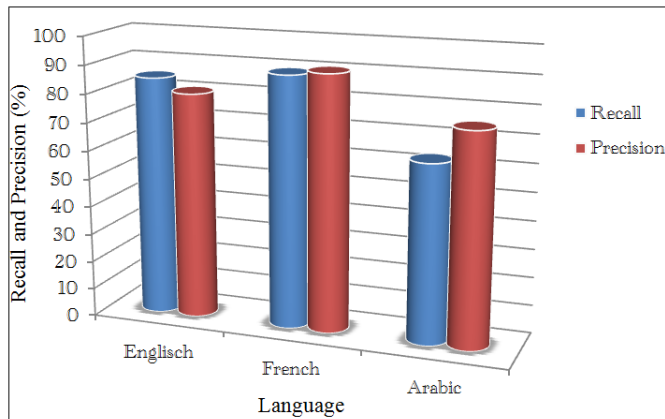
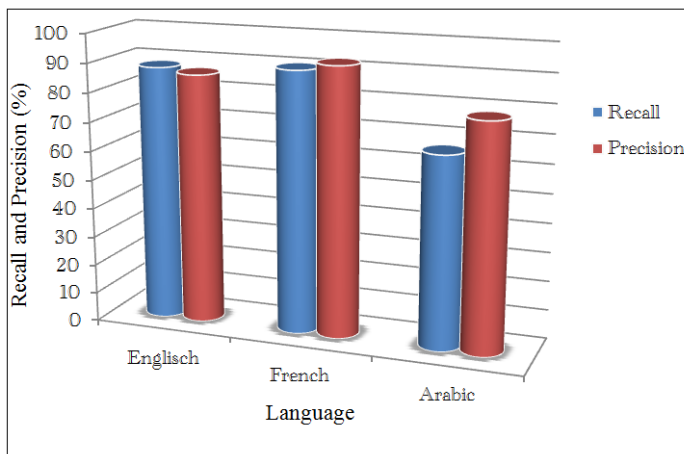


Figure 12. Recall and Precision performance for high-level lingware applications



the same. This similarity is explained by the fact that these languages share a lexical similarity (similarity in both form and meaning). Indeed, they have the same alphabet. They sometimes use similar grammatical structures and have several words in common. However, the overall means of the precision associated with these languages (86.3% and 92.3%) are better than the overall mean of the precision associated with the Arabic language (78%). This gap is explained by the fact that the Arabic language differs morphologically, syntactically and semantically from the English and French languages. For instance, syntactically, Arabic sentences are long with complex syntax and its components can be swapped without affecting the structure or meaning. These issues lead to a syntactic and semantic ambiguity. Besides, the NLP toolkits and frameworks used to propose the LingOnto are more mature for English and French Languages than Arabic language. Furthermore, this gap affects the performance of LingFramework in identifying valid pipelines for high-level Arabic applications as shown in Figure 12. This is explained by the fact that, the high-level applications depend on the low-level ones. For instance, syntactic analysis like parsing usually requires words to be clearly delineated and part-of-speech tagging or morphological analysis to be performed first. This means, in practice, that texts must be tokenized, their sentences clearly separated from each other, and their morphological properties analyzed before beginning the parsing process.

## 6.2. 2.Evaluation of the Usability

The authors evaluate the usability of LingFramework using the standard System Usability Scale (SUS) questionnaire (Brooke, 2013). Fifteen participants enter this study. They are expert (i.e., linguistic, engineering and/or ontology experts) and novice users (i.e., non-linguistic, non-engineering and/or non-ontology experts).

**Table 2. The overall results of the SUS questionnaire**

SUS Subjective Feedback	LingFramework
I think that I would like to use this system frequently	4,75
I found the system unnecessarily complex	1,37
I thought the system was easy to use	4,87
I think that I would need the support of a technical person to be able to use this system	1
I found the various functions in this system were well integrated	4,5
I thought there was too much inconsistency in this system	1
I would imagine that most people would learn to use this system very quickly	4,37
I found the system very cumbersome to use	1
I felt very confident using the system	4
I needed to learn a lot of things before I could get going with this system	1,5
SUS Score	28,36

All the participants filled out a questionnaire to describe their opinion about the usability of LingFramework. The authors use the SUS questionnaire for the reason that it is a well-known and widely used method for the measurement of the user's perception of the usability evaluation of systems. The first step in scoring a SUS is to determine each item's score contribution, which will range from 0 to 4. For positively-worded items (odd numbers), the score contribution is the scale position minus 1. For negatively-worded items (even numbers), the score contribution is 5 minus the scale position. To get the overall SUS score, the authors multiply the sum of all item score contributions by 2.5, which produces a score that can range from 0 (very poor perceived usability) to 100 (excellent perceived usability).

According to (Bangor et al., 2009), a SUS score above 68 is considered above average. Table 2 shows the results of the SUS questionnaire.

The SUS scores for LingFramework used by the expert users (82.5 (33\* 2.5)) and the novice ones (75 (30\* 2.5)) indicates a high level of usability.

## 7. CONCLUSION

This paper addresses the issue of assisting the user in understanding the different aspects of the linguistic domain and easing the process of proposing lingware applications. The authors propose the LingFramework which is a smart framework for handling multilingual linguistic knowledge. It targets not only expert users, but also novice ones. It is based on the LingOnto ontology. Compared to available works, this multilingual linguistic domain ontology allows representing linguistic data, linguistic processing functionalities and linguistic processing features. Furthermore, it allows reasoning, via a SWRL based reasoning engine, about the aforementioned knowledge. Currently, three languages are supported: English, French and Arabic. LingOnto is designed to be used mainly by linguistic users, who are usually not familiar with ontologies. To attempt this issue, the authors propose the LingGraph



user friendly ontology visualization tool. It is designed to be used by both ontology and non-ontology expert users. To support an understandable visualization, LingGraph is based on a “smart” search functionality that relies on a SPARQL pattern-based approach. This latter extracts and visualizes an excerpt ontological view from LingOnto containing only components corresponding to the user’s needs. Finally, the authors evaluate (1) the performance of LingFramework in identifying valid NLP pipelines for 63 proposed lingware applications and (2) the usability of its user interfaces. The results show that the proposed framework is useful in identifying valid NLP pipelines and it is easy to use.

For future research, the authors plan to extend LingOnto by giving the possibility to linguistic experts adding new linguistic knowledge concepts and their associated object properties. Moreover, they suggest exploiting the NLP domain expert’s feedback to improve the Not Valid identified NLP pipelines. In addition, they plan to execute the valid pipelines by discovering concrete linguistic web services that match each required linguistic processing functionality in the pipeline. Besides, they plan to allow the LingOnto ontology to be referenced by the Linked Open Vocabularies (LOV) platform. Finally they plan to extend LingFramework to handle the natural language data imperfections which represents actually a crucial issue in the NLP field.

## REFERENCES

- A. (2004). Semantic Web Rule Language Combining OWL and RuleML.
- Achich, N., Algergawy, A., Bouaziz, B., & König-Ries, B. (2018). BioOntoVis: An Ontology Visualization Tool. In *EKAW*, (pp. 49–52). Posters & Demos.
- Alani, H. (2003). TGVizTab: An ontology visualisation extension for Protégé. *Proc. of the 2nd Workshop on Visualizing Information in Knowledge Engineering (VIKE 03)*.
- Antoniou, G., & Van Harmelen, F. (2004). Web ontology language: Owl. In *Handbook on ontologies*, (pp. 67–92). Springer. doi:10.1007/978-3-540-24750-0\_4
- Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., & Padró, M. (2006). FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In *LREC*. 6, pp. 48–55).
- Baklouti, N., Bouaziz, S., Gargouri, B., Aloulou, C., & Jmael, M. (2010). Towards the reuse of lingware systems: A proposed approach with a practical experiment. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, (pp. 566–572). doi:10.1145/1967486.1967573
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114–123.
- Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., & Gorrell, G. (2013). GATE Teamware: A web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4), 1007–1029. doi:10.1007/s10579-013-9215-6
- Brooke, J. (2013). SUS: A retrospective. *Journal of Usability Studies*, 8(2), 29–40.
- Catenazzi, N., Sommaruga, L., & Mazza, R. (2009). User-friendly ontology editing and visualization tools: the OWLeasyViz approach. In *13th International Conference Information Visualisation*, (pp. 283–288). doi:10.1109/IV.2009.34
- Chiarcos, C., & Hellmann, S. (2017). OnLiT: An Ontology for Linguistic Terminology. In *Language, Data, and Knowledge. Proceedings*, 10318, 42.
- Chiarcos, C., & Sukhareva, M. (2015). OLiA—ontologies of linguistic annotation. *Semantic Web*, 6(4), 379–386. doi:10.3233/SW-140167
- Citeseer. Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, (pp. 336–343). IEEE.
- Costea, E.-A. (2020). Machine Learning-based Natural Language Processing Algorithms and Electronic Health Records Data. In *Linguistic and Philosophical Investigations*, 93–99.
- de Cea, G. A., de Mon, I., Gomez-Perez, A., & Pareja-Lora, A. (2004). OntoTag’s linguistic ontologies: improving semantic Web annotations for a better language understanding in machines. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC*, 2, pp. 124–128. IEEE. doi:10.1109/ITCC.2004.1286605
- Fairen-Jimenez, D., Moggach, S., Wharmby, M., Wright, P., Parsons, S., & Duren, T. (2011). Opening the gate: Framework flexibility in ZIF-8 explored by experiments and simulations. *Journal of the American Chemical Society*, 133(23), 8900–8902. doi:10.1021/ja202154j PMID:21553843
- Farrar, S. & Langendoen, D. (2010). An OWL-DL implementation of GOLD: An ontology for the Semantic Web. *Linguistic Modeling of Information and Markup Languages: Contributions to Language Technology*.
- Florence, M. (2021). MLGrafViz: Multilingual ontology visualization plug-in for protégé. *Computer Science and Information Technology*, 2(1), 43–48.
- Gangemi, A., Guarino, N., Masolo, C., & Oltramari, A. (2010). Interfacing wordnet with dolce: towards ontowordnet. *Ontology and the Lexicon: A Natural Language Processing Perspective*, 36–52. Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6), 907–928.

- Gangemi, A., Guarino, N., Oltramari, A., & Borgo, S. (2002). Cleaning-up WordNet's top-level. In *Proceedings of the 1st International WordNet Conference*, (pp. 21–25).
- Haddar, K., & Hamadou, A. B. (2009). An Ellipsis Resolution System for the Arabic Language. *Int. J. Comput. Process. Orient. Lang.*, 22(4), 359–380. doi:10.1142/S1793840609002159
- Hanke, H., & Knees, D. (2017). A phase-field damage model based on evolving microstructure. *Asymptotic Analysis*, 101(3), 149–180. doi:10.3233/ASY-161396
- Hayashi, Y., & Narawa, C. (2012). Classifying Standard Linguistic Processing Functionalities based on Fundamental Data Operation Types. In LREC, (pp. 1169–1173).
- Henrich, V. & Hinrichs, E. (2010). GernEdiT: a graphical tool for germanet development. In *Proceedings of the ACL 2010 System Demonstrations*, (pp. 19–24).
- Holten, D., & Van Wijk, J. J. (2009). Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3), 983–990. doi:10.1111/j.1467-8659.2009.01450.x
- Hussain, A., Latif, K., Rextin, A. T., Hayat, A., & Alam, M. (2014). Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1), 355–367. doi:10.12785/amis/080145
- Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., DiPersio, D., Shi, C., Suderman, K., Verhagen, M., Wang, D., & Wright, J. (2015). The language application grid. In *International Workshop on Worldwide Language Service Infrastructure*, (pp. 51–70).
- Ide, N., & Romary, L. (2004). A registry of standard data categories for linguistic annotation. In *4th International Conference on Language Resources and Evaluation-LREC,04*, (pp. 135–138).
- Ishida, T. (2006). Language grid: An infrastructure for intercultural collaboration. In *International Symposium on Applications and the Internet (SAINT'06)*, (p.5–pp). doi:10.1109/SAINT.2006.40
- Jarrar, M. (2021). The Arabic ontology—an Arabic wordnet with ontologically clean content. *Applied Ontology*, 16(1), 1–26. doi:10.3233/AO-200241
- Klimek, B., Ackermann, M., Brümmer, M., & Hellmann, S. (2021). MMoOn Core—the Multilingual Morpheme Ontology. *Semantic Web*, 12(5), 813–841. doi:10.3233/SW-200412
- Klingström, T., Hernández-de-Diego, R., Collard, T., & Bongcam-Rudloff, E. (2017). Galaksio, a user friendly workflow-centric front end for Galaxy. *EMBnet.Journal*, 23(0), 897. doi:10.14806/ej.23.0.897
- Konchady, M. (2008). *Building Search Applications: Lucene, LingPipe, and Gate*. Lulu. com.
- Krivov, S., Williams, R., & Villa, F. (2007). GrOWL: A tool for visualization and editing of OWL ontologies. *Journal of Web Semantics*, 5(2), 54–57. doi:10.1016/j.websem.2007.03.005
- Laatar, R., Rhayem, A., Aloulou, C., & Belguith, L. H. (2022). Towards a Historical Ontology for Arabic Language. *Investigation and Future Directions. In International Conference on Intelligent Systems Design and Applications*, (pp. 1078-1087). Springer, Cham.
- Lanzenberger, M., Sampson, J., & Rester, M. (2009). Visualization in Ontology Tools. In *International Conference on Complex, Intelligent and Software Intensive Systems*, (pp. 705–711). doi:10.1109/CISIS.2009.178
- Lefever, E. (2016). A hybrid approach to domain-independent taxonomy learning. *Applied Ontology*, 11(3), 255–278. doi:10.3233/AO-160170
- Liebig, T., & Noppens, O. (2005). OntoTrack: A semantic approach for ontology authoring. *Journal of Web Semantics*, 3(2-3), 116–131. doi:10.1016/j.websem.2005.06.004
- Lohmann, S., Negru, S., Haag, F., & Ertl, T. (2014). VOWL 2: User-oriented visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, (pp. 266–281). doi:10.1007/978-3-319-13704-9\_21
- Lohmann, S., Negru, S., Haag, F., & Ertl, T. (2016). Visualizing ontologies with VOWL. *Semantic Web*, 7(4), 399–419. doi:10.3233/SW-150200
- Loos, E. E. (2004). *Glossary of linguistic terms*. SIL International.

- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, (pp. 55–60). doi:10.3115/v1/P14-5010
- Meltzer, P. S., Kallioniemi, A., & Trent, J. M. (2002). Chromosome alterations in human solid tumors. In B. Vogelstein and K.W. Kinzler (Eds.), *The Genetic Basis of Human Cancer*, (pp. 93–113).
- Mohanam, M., & Samuel, P. (2016). Open NLP based refinement of software requirements. *International Journal of Computer Information Systems and Industrial Management Applications*, 8, 293–300.
- Murray, P. R., Rosenthal, K. S., Kobayashi, G. S., & Pfaller, M. A. (2002). *Medical Microbiology*, (4th ed.). Mosby.
- Neji, M., Ghorbel, F., & Gargouri, B. (2019). Visualizing a Linguistic Ontology with Ling-Graph. In *11th International Conference, ICCCI*, 11683, (pp. 41–52).
- Peroni, S., Motta, E., & d'Aquin, M. (2008). Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In *Asian Semantic Web Conference*, (pp. 242–256). doi:10.1007/978-3-540-89704-0\_17
- Schalley, A. C. (2019). Ontologies and ontological methods in linguistics. *Language and Linguistics Compass*, 13(11), e12356. doi:10.1111/lnc3.12356
- Schuurman, I., Windhouwer, M., Ohren, O., & Zeman, D. (2016). CLARIN concept registry: the new semantic registry. In *Selected Papers from the CLARIN Annual Conference*, Poland, (pp. 62–70).
- Singh, G., Prabhakar, T., Chatterjee, J., Patil, V., & Ninomiya, S. (2006). OntoViz: visualizing ontologies and thesauri using layout algorithms. In the *fifth international conference of the Asian Federation for Information Technology in Agriculture (AFITA 2006)*.
- Su, L. T. (1994). The relevance of recall and precision in user evaluation. *Journal of the American Society for Information Science*, 45(3), 207–217.
- Wilcock, G. (2009). Unstructured Information Management Architecture (UIMA). In *2nd UIMA@ GSCL Workshop*.
- Wilson, E. (1991). Active vibration analysis of thin-walled beams. [PhD thesis] University of Virginia.
- Witt, A. W., & Metzger, D. Dordrecht, Springer. Fellbaum, C. (1998). WordNet: An electronic lexical database. MIT Press.
- Zhou, M., Duan, N., Liu, S., & Shum, H.-Y. (2020). Progress in neural NLP: Modeling, learning, and reasoning. *Engineering*, 6(3), 275–290.
- Ziad, H., McCrae, J. P., & Buitelaar, P. (2018). Teanga: a linked data based platform for natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

## ENDNOTES

- <sup>1</sup> <https://oa.upm.es/13827/>
- <sup>2</sup> <https://protegewiki.stanford.edu/wiki/SOVA>
- <sup>3</sup> <https://protegewiki.stanford.edu/wiki/OWLViz>
- <sup>4</sup> <https://github.com/mariemNeji/LingOnto>
- <sup>5</sup> <https://github.com/mariemNeji/Ling-Graph>.

*Mariem Neji is currently pursuing a PhD degree with the Faculty of Economics and Management of Sfax (FSEGS)-University of Sfax (Tunisia). Mariem Neji conduct her research within the Natural language Processing Research Group (ANLP-RG) of Multimedia, Information systems and Advanced Computing Laboratory (MIRACL). Her main research interests include application to Natural Language Processing*

*Bilel Gargouri is an associate Professor in Computer Science at the University of Sfax - Tunisia. He obtained the habilitation to supervise research activities in 2016. Currently, he works on the standardization of lexical resources and Digital Humanities. He has contributed to the development of the LMF - ISO 24613 standard as an expert of the Arabic language. Mr GARGOURI has participated in the organization of several scientific events and acts as a program committee member for several national and international conferences. Moreover, he has reviewed many papers for international peer-reviewed journals. He was the director of the IT department at the Faculty of Economics and Management sciences of Sfax from 2014 to 2020. Actually, he is the Director of Studies and the Vice-Dean.*

*Nada Mimouni is lecturer in computer science at the Conservatoire National des Arts et Métiers (CNAM) of Paris since 2019. Her research interest includes knowledge extraction and representation and the Semantic Web with the use of automatic language processing methods for data analysis and exploration. Before joining CNAM, she was an assistant professor in Paris Sorbonne University and participated in several projects as a postdoctoral researcher at Télécom Paris and Paris Dauphine University in the framework of the Interdisciplinary and Strategic Research Initiatives (IRIS) led by Paris Sciences et Lettres (PSL) Research University. She obtained her Ph.D. from Paris Nord University in 2015. She received the engineering degree in computer science from the University of Tunis and the M.S. degree from Nancy II University in 2008.*

*Elisabeth Métais is a Full Professor in Computer Science at the CNAM University (Paris, France) and a researcher in the CEDRIC Laboratory since September 2000. Her research initially concerned Data Bases Design and she has been interested since the early nineties in applying Artificial Intelligence and Natural Language Processing techniques to Database Design. In 1995 she initiated the NLDB conference in Versailles (France). Since the 2000s she has turned to medical computing, through projects such as Data Warehouses in resuscitation, or the design of Serious Games for therapeutic purposes in urology. She is currently interested in memory prosthesis for elderly persons and Alzheimer disease patients, based on the Semantic Web technologies and has recently managed a project concerning the help of ICTs and smart cities for Alzheimer patients living in the City of Paris.*