An Ontology based Smart Management of Linguistic Knowledge

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

An Ontology based Smart Management of Linguistic Knowledge

Mariem Neji[1*], Fatma Ghorbel[1,2†], Bilel Gargourir[1†], Nada Mimouni[2†] and Elisabeth Métais[2†]

[1*]Miracl Laboratory, University of Sfax, Faculty of Economics and Management of Sfax, 3038, Sfax, Tunisia.
[2]Cedric Laboratory, Centre d'études et de recherche en informatique et communications, 75003, Paris, France.

*Corresponding author(s). E-mail(s): mariem.neji@yahoo.fr;
Contributing authors: fatmaghorbel6@gmail.com;
bilel.gargouri@fsegs.rnu.tn ; nada.mimouni@lecnam.net;
elisabeth.metais@cnam.fr;
†These authors contributed equally to this work.

Abstract

Natural language processing provides a very significant contribution to various application areas such as multilingual big data, information retrieval, data integration and multilingual web. However, handling linguistic knowledge to develop such lingware applications is a crucial issue, especially for linguistic novice users. To deal with this issue, a "smart" linguistic knowledge management may help the user to understand the meaning, scope and especially the use of related techniques and algorithms. In this paper, we propose (1) a semantic processing of linguistic knowledge based on a multilingual linguistic domain ontology, called LingOnto. Compared to related work, LingOnto does not only handles 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. This is particularly useful for novice users, but can also provide new perspectives for expert ones. LingOnto covers the French, English and Arabic languages (2) an assisted visualization based on a user friendly ontology visualization tool called LingGraph to facilitate the interaction with LingOnto. 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 to visualize only an ontological view based on the user's needs and preferences. In order to evaluate LingOnto and measure its efficiency, we applied it to a framework of identifying valid natural language processing pipelines. Finally, we give the results of the carried-out experiments.

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

1 Introduction

The importance of linguistic knowledge is increasing in many application areas, such as multilingual big data [1], [2], information retrieval [3], question-answering and NLP-based applications [4], data integration [5], multilingual web [6], among others.

However, handling linguistic knowledge to develop such lingware applications is a crucial issue. To deal with this issue, a smart linguistic knowledge management may help the user to understand the meaning, scope and especially the use of linguistic knowledge. 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 [7]. Besides, they do not support multilingualism. Ontologies were proven to be more useful as they provide more precise and semantically richer results [5]. 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, we present a "smart" management of linguistic knowledge. To this end, we propose a multilingual ontology called LingOnto, that covers the different aspects of the NLP domain. It aims to make a wide range of linguistic data, linguistic processing functionalities and linguistic processing features easily accessible to the user. 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, we 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.

In order to evaluate LingOnto, we experiment it in the context of Lingware engineering. Particularly, it is applied to a framework of identifying valid NLP pipelines.

The current paper is organized as follows. Section 2 presents some related works. Section 3 presents the multilingual linguistic domain ontology LingOnto. Section 4 presents the proposed user friendly ontology visualization tool LingGraph. The evaluations of the performance of LingOnto will be presented in Section 5. Finally, Section 6 draws conclusions and future research directions.

2 Related Work

The present work is closely related to the following research areas: *(1)* linguistic knowledge representation and *(2)* ontology visualization.

2.1 Linguistic Knowledge Representation

Various approaches focusing on linguistic knowledge representation are proposed. We distinguish two main categories: *(1)* registries-based approaches and *(2)* ontologies-based approaches.

2.1.1 Registries-Based Approaches

The SIL glossary of linguistic terms [8] 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 [9].

The ISOcat data category registry [10] 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 [11] takes over the work of the ISOcat data category registry. However, this latter still provides very limited structural and relational information [11].

We 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*" [9].

2.1.2 Ontologies-Based Approaches

In [12], 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 [12]. Moreover, it provides a poor classification of the types of numbers [13].

In [14], 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 [13]. The development of this ontology was stopped in 2010.

In [15], 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[1] ontologies proposed by [16]. One important difference is that the OntoTag ontologies are considering only the languages of the Iberian peninsula (in particular Spanish).

In [13], 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.

We 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

[1] http://oa.upm.es/13827/

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 experts and they overlook the importance of the usability and understandability requirements. According to [17], the generated visualizations "are hard to read for casual users". For instance, GrOWL and SOVA [2] 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 [3], OntoTrack [18], KC-Viz and OntoViz show only specific element(s) of the ontology. For instance, the OWLViz and KC-Viz visualize only the class hierarchy of the ontology and OntoViz shows only inheritance relationships between the graph nodes. This is different with TGViz Tab [19] and NavigOWL [20] 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 [21], Protégé VOWL [17] and WebVOWL [17]. 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 LingOnto: A Multilingual Linguistic Domain Ontology

In this section, we present our ontology based smart management of linguistic Knowledge called LingOnto. It is freely available online [4]. 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

[2] http://protegewiki.stanford.edu/wiki/SOVA
[3] http://protegewiki.stanford.edu/wiki/OWLViz
[4] https://github.com/mariemNeji/LingOnto

via a reasoning engine, inferring new linguistic knowledge from those initially entered and assisting in the process of proposing lingware applications (e.g., it helps the user to avoid incoherency errors by assisting him selecting only compatible linguistic processing functionalities.).

3.1 Representing Linguistic Knowledge

We are based on the design principles defined by [22], which are objective criteria for proposing and evaluating ontology designs, such as clarity, coherence, minimal encoding bias and minimal ontological commitments. Following these principles, we 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.

3.1.1 Linguistic Data Classification

Referring to the ISOcat standard, we identify a set of linguistic data concepts. We choose this registry for the following reasons:

- It covers more terms of linguistic data categories compared to other resources. For instance, it holds 115 possible values of "*PartOfSpeech*" such as *(Adjectif)*, *(Verb)*, *(Noun)* and *(Adverb)* while; the Gold ontology has only 81 values.
- It defines linguistic data categories at several levels such as syntactic, morphosyntactic, terminological and lexical.
- It supports various languages. For instance, it provides description of usage in language-specific contexts, including definitions, usage notes and/or lists of values.

For each extracted linguistic data concept, we identify the concepts which are related to it as well as the names of the associated relations. Fig. 1 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, we 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, we 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).

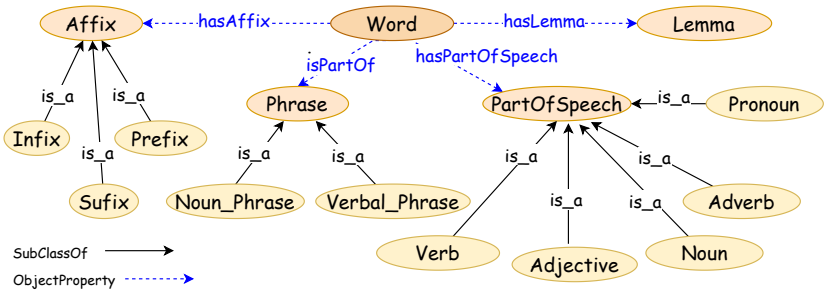

Fig. 1 The classification of some Arabic linguistic data

3.1.2 Linguistic Processing Functionalities Classification

Referring to well-known NLP toolkits such as Apache OpenNLP [23], StandfordCoreNlP [24], FreeLing [25] and LingPipe [26] and two language processing platforms which are Language Grid [27] and Gate [28], we identify a set of linguistic processors such as *POS Tagger*, *Lemmatizer*, *Morphological Analyzer* and *Chunker*. Some of these linguistic processors implements often one or two linguistic processing functionalities. For instance, a *Morphological Analyzer* processor for French and English languages usually implement *Paragraph splitting*, *Sentence splitting*, *Tokenization*, *POS tagging* and *Lemmatization* processing functionalities. Nerveless, a *Morphological Analyzer* processor for Arabic language, especially for analysing undiacritized texts, implements *Paragraph splitting*, *Sentence splitting*, *Tokenization*, *Diacritization*, *POS tagging* and *Lemmatization* processing 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 [29], an hierarchical inter-dependencies between the linguistic processing functionalities exists. 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, we identify the object property *"Requires"*. As shown in Fig. 2, the ("*Tokenization*") class is in relation with the ("*Sentence_Splitting*") class through the object property *"Requires"*. Moreover, each linguistic processing functionality uses various linguistic data as inputs and others as outputs. Hence, we propose the objects properties *"Has_Input"* and *"Has_Output"*. For instance, as shown in Fig. 2,

the (*"Tokenization"*) class is in relation with the (*"Sentence"*) class through *"Has_Input"* object property. It is also in relation with the (*" Word"*) class through *"Has_Output"* object property.

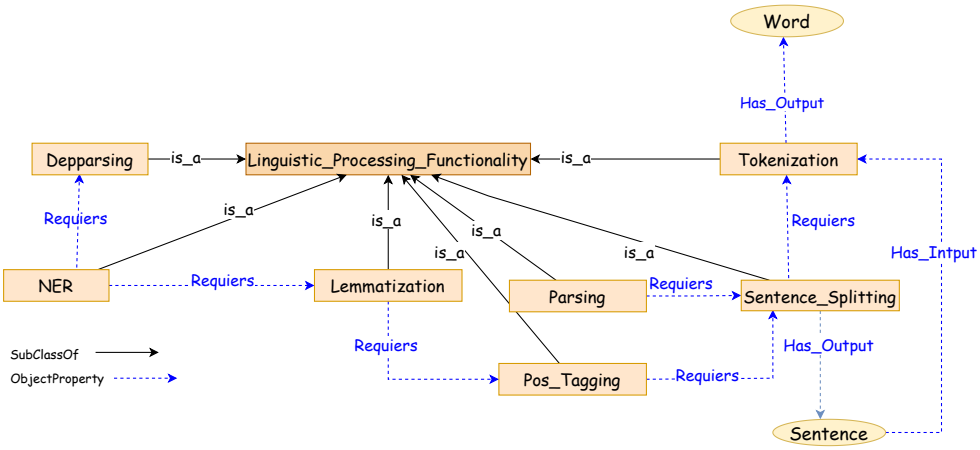

Fig. 2 The classification of some Arabic linguistic processing functionalities

3.1.3 Linguistic Processing 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. We present in Table. 1 some examples of the linguistic processing features.

Table 1 Examples of Linguistic Processing Features.

Linguistic Processing Features	Examples
Processing Level	Lexical, Morphological, Syntactic,and Semantic
Phenomenon	Ellipsis, Accord, and Anaphora
Analysis Type	Structural, Thematic, Syntagmatic,Top-down Bottom-Up, Profound, and Surfacing or Chunking
Approach	Linguistic, Statistic, and Hybrid
Formalism	Unification Grammar and Resolution Algorithm
Resource	WordNet-LMF and GermaNet
Language	English, Arabic, and French
Treatment Type	Analysis, Generation, and Hybrid

The English, French and Arabic languages are based on the same linguistic processing features. Indeed, according to [30], 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.

Fig. 3 shows the proposed classification of the linguistic processing features. Each processing level is characterized by its related phenomena. Hence, we define the object property *"has_Phenomenon"* between *("Processing_ Level")* 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, we define the *"refined_into"* reflexive object property. The linguistic phenomenon has also the relations *"supported_By"*and *"treated_By"*, 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 to ward its antecedent *"the plate"*. Finally, each processing level uses a linguistic resource related to a phenomenon. Hence, we define the object property *"has_Resource"* relating the *("Processing_ Level")* and *("Linguistic_Resource")* classes.

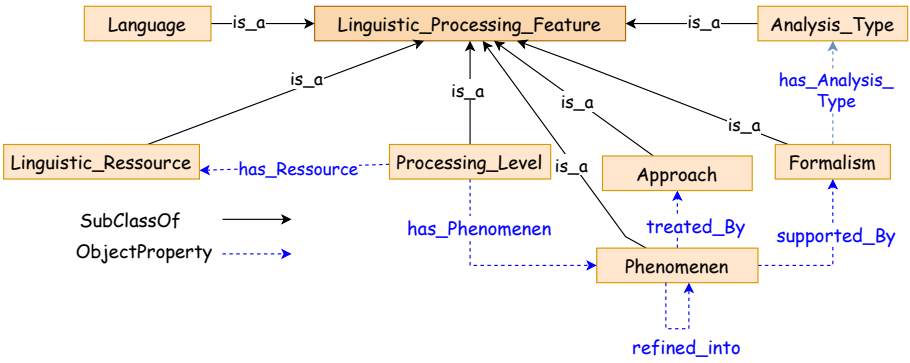

Fig. 3 The classification of some linguistic processing features

3.2 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. We categorize the proposed SWRL rules into two categories: *(1)* SWRL rules for lingware applications development assistant and *(2)* SWRL rules for NLP domain understanding assistant.

3.2.1 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. Fig. 4 shows some examples.

- 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".

3.2.2 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. Fig. 5 shows some examples.

- 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.

4 LingGraph: Ontology Visualization Tool of LingOnto

The LingOnto domain ontology is designed to be used by users, who are not necessary ontology experts. Visualizations are usually proposed to help in this regard by assisting in the sense-making. Moreover, the large amount of linguistic knowledge covered by LingOnto makes the visualization hard to comprehend due to the visual clutter and information overload. To overcome this issue, we propose a user friendly ontology visualization tool called LingGraph. The main aim of this tool is to offer an understandable visualization to both ontology and non-ontology expert users. To support the large amount of linguistic knowledge covered by LingOnto, 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

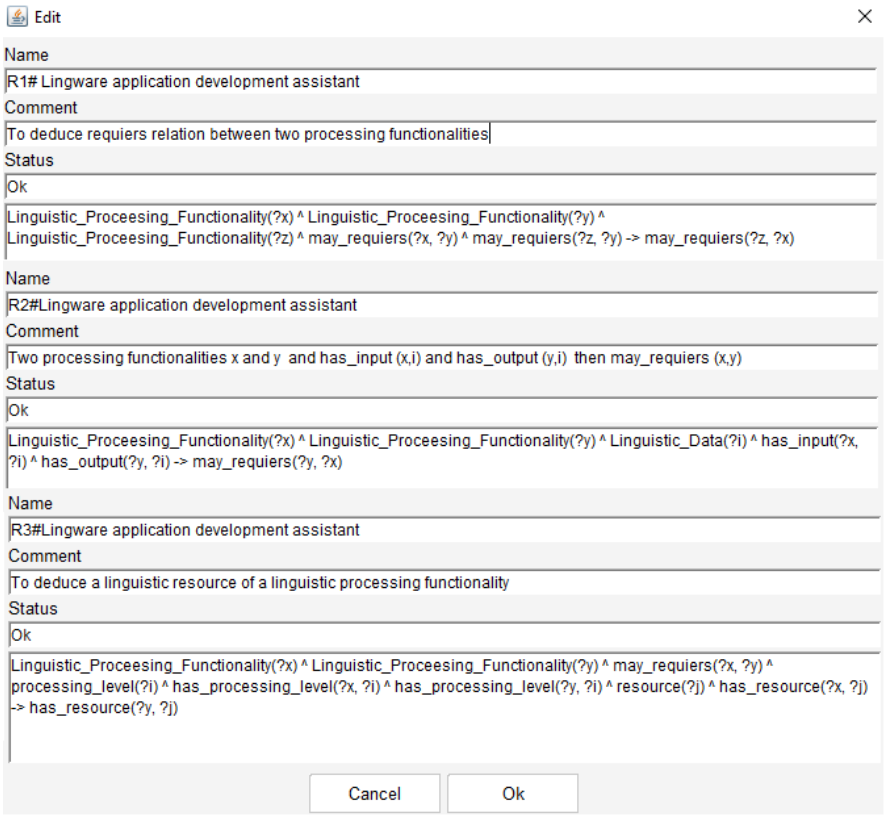

Fig. 4 Example of SWRL rules for lingware applications development assistant

corresponding to the user's needs. Moreover, it offers an easy-to-understand wording. For instance, it does not use a semantic web vocabulary. LingGraph is mainly designed to be integrated into a linguistic framework. It can be integrated into other applications for non-ontology experts and it can be used as a standalone application by ontology experts.

4.1 Graph-based visualization

LingGraph visualizes the ontology, formalized in OWL2 as a graph. It is based on a force field algorithm. This latter has two main advantages. (1) It ensures an optimal use of the screen. It displays the nodes in a way that those that are closely connected are shown in the center of the visualization, while the ones that are less connected are placed at the edges. (2) It improves the readability of the graph, by avoiding crossing links and displaying all the key elements of the ontology. Moreover, it allows representing the object properties between the concerned nodes by using labeled links. In order to be differentiated from the instances, the classes are displayed in a larger size.

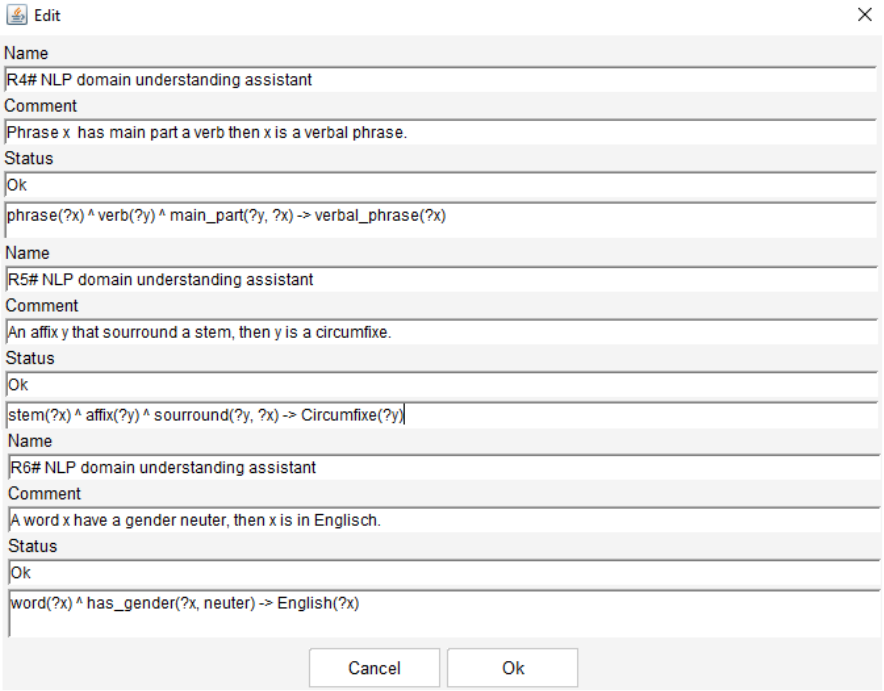

Fig. 5 Example of SWRL rules for NLP domain understanding assistant.

4.2 Smart Search Interaction Functionality

The smart search interaction functionality is based on a SPARQL pattern-based approach. The aim is to extract and visualize an excerpt ontological view, from LingOnto, which contains only components corresponding to users need's. This latter is materialized by a set of predefined search criteria C = (C_1,...,C_n) such as *"Abstraction Level"*, *"Processing Level"* and *"Language"*. For each criterion C_i ($i \in [1, n]$), a set of preferences CP = ($CP_{i/1}$,...,$CP_{i/m}$) is associated. For example, the preferences associated with the criterion *"Processing Level"* are: *("lexical level")*, *("morphological level")*, *("semantic level")* and *("syntactic level")*. The user selects more than one preference of each criterion.

We asked some users (expert and novice users) to fill a pre-questionnaire about what they need to know as linguistic knowledge. We notice that their needs are very regular as all of them search the abstraction level (e.g., linguistic data and/or processing functionalities and features) of a given processing level(s) or/and a given language(s). This observation leads us to propose an approach based on a set of SPARQL patterns P = (P_1,...,P_k).

4.2.1 Pattern Definition

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

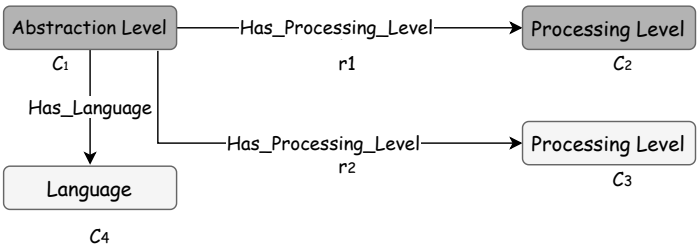

Fig. 6 An Example of a pattern

- G is a connected RDF graph, which describes the general structure of the pattern and represents a family of queries;
- 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.

Fig. 6 displays a 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 vertexes 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 a resource, in order to turn the pattern into a query. This means that, to have 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.

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

- q is a class and α 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 α 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 α.

- q is a property and α 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 α.

The instantiation of the pattern shown in Fig. 6 leads, after substitution of each qualifying element by the selected preferences, to the query graph shown in Fig. 7.

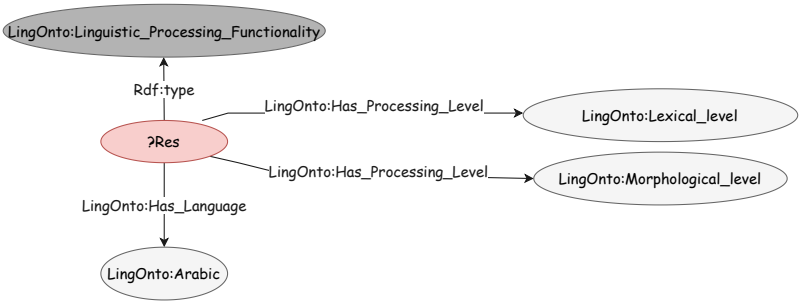

Fig. 7 Query graph resulting from the instantiation of the pattern of Fig. 6

4.2.3 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, we 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 we 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.

We 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. Fig. 8 shows the generated SPARQL query corresponding to the query graph in Fig. 7.

5 Experimentation

We apply the proposed ontology LingOnto to a linguistic framework of identifying valid linguistic NLP pipelines. To ensure an understandable visualization of LingOnto, we integrate to this framework our ontology visualization tool LingGraph. Then, we evaluate the efficiency of our ontology in identifying valid NLP pipelines.

SPARQL query:

```
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
               }
```

Fig. 8 The generated SPARQL query associated to the request graph shown in Fig. 7

5.1 Application to an NLP Pipelines Identification Framework

Lingware applications are defined as a sequence of many individual components to solve real-world problems [31]. 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. The LingOnto is applied to a framework of identifying valid NLP pipelines. It targets novice users in the lingware engineering area.

As shown in Fig. 9, 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.

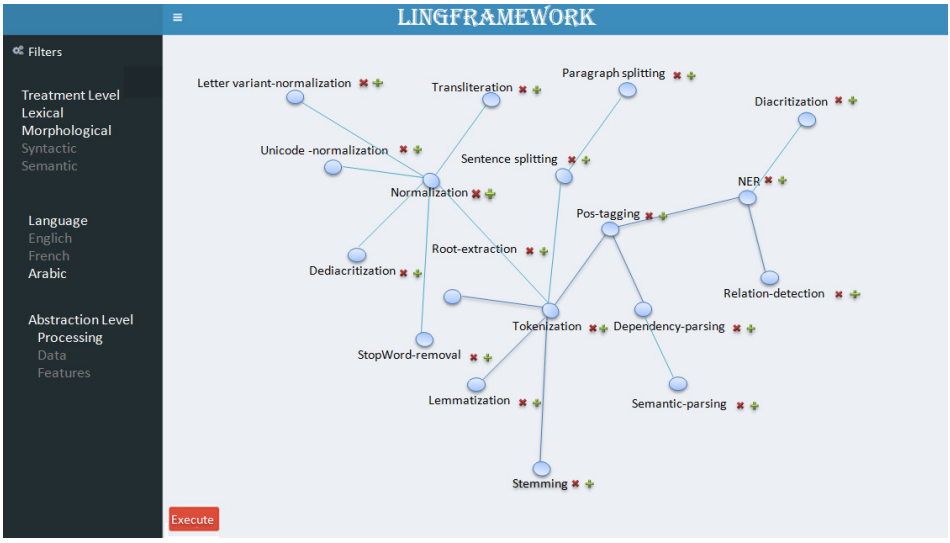

Fig. 9 Ontological view generation screenshot

Then, the user starts the process of identifying an NLP pipeline related to the target lingware application. Consequently, the framework 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, Fig. 10 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.

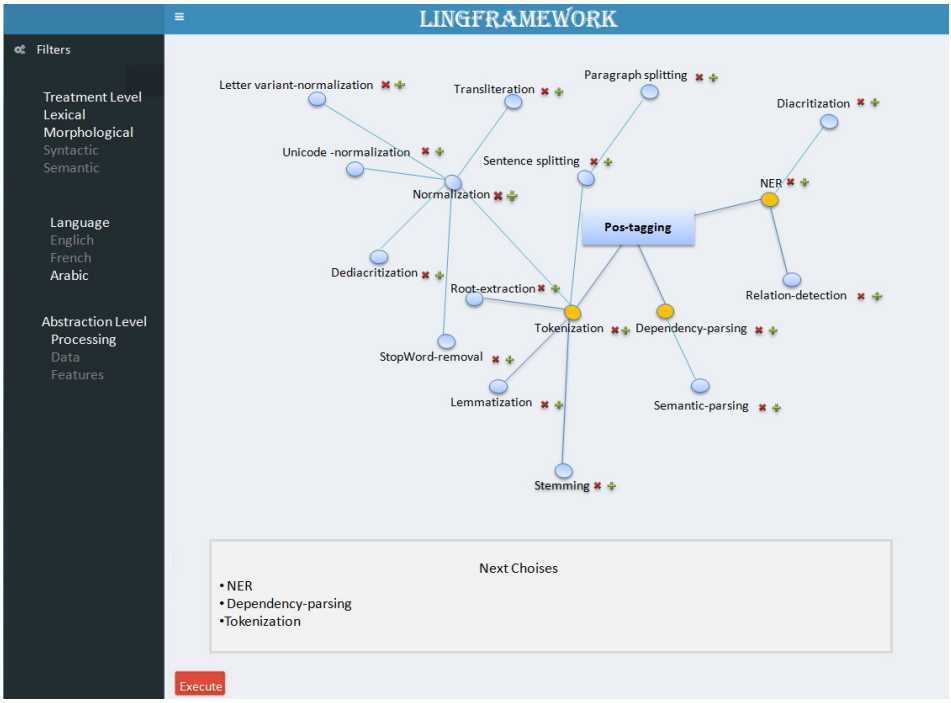

Fig. 10 NLP pipeline construction screenshot

If the user selects a processing functionality out of the list under *"Next Choices"*, the framework displays an error message *"Incompatible Functionalities"* and indicates using the red color an alternative valid pipeline. As shown in Fig. 11, the *("Diacritization")* functionality can be added to the pipeline only after *("Pos_tagging")* and *("NER")* functionalities. The final NLP pipeline is shown in Fig. 12.

5.2 Evaluation

In this section, we evaluate the performance of LingOnto in identifying valid NLP pipelines associated to lingware applications. This evaluation consists of three steps:

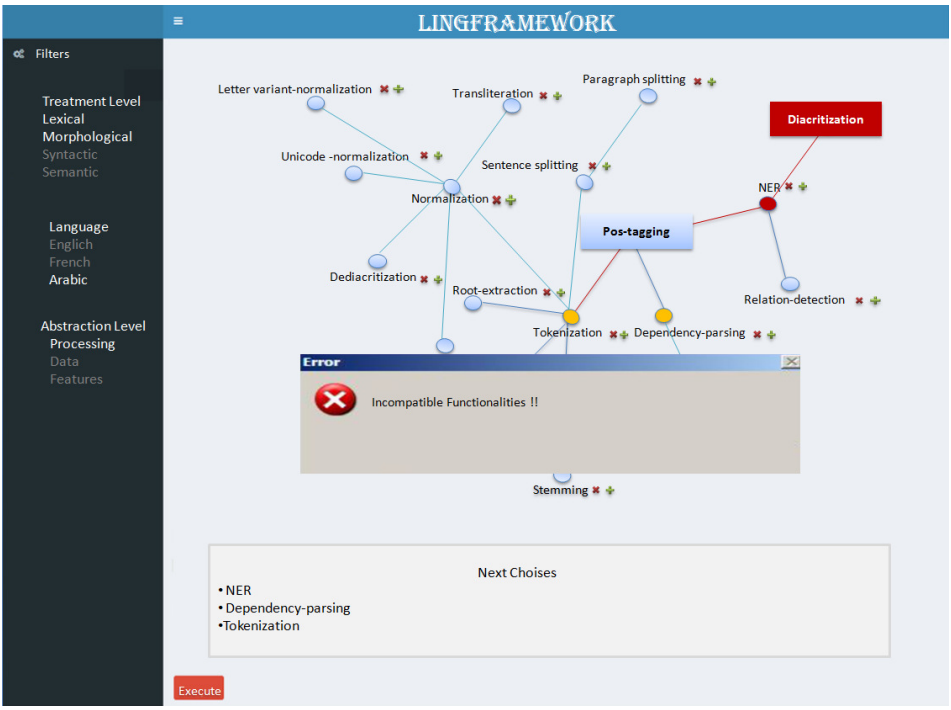

Fig. 11 Alternative NLP pipeline proposition screenshot

- **Step 1**: we propose 63 lingware applications, which have to be solved by identifying their corresponding NLP pipelines using LingOnto. We classify these applications into *(1)* low level and *(2)* high level applications. Then, we classify applications in each group according to the language (i.e., French, Englich and Arabic). Table. 2 shows some examples.

Table 2 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 analysis 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**: we recruit three linguistic experts. The first one is a member of the Arabic Natural Language Processing Research Group (ANLP-RG) of MIR-ACL 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). We ask each expert to provide, manually,

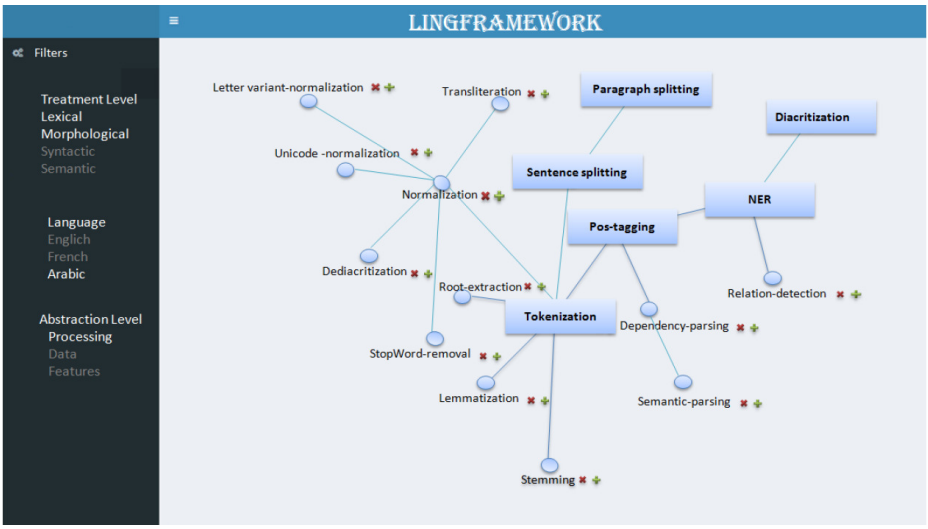

Fig. 12 The final NLP pipeline

all the possible pipeline(s) which may solve each lingware application related to their native language (i.e., French, English and Arabic).

- **Step 3**: we identify, using the linguistic framework, 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.

We use the precision and recall metrics [32] to evaluate the performance of LingOnto. The recall measures the proportion of valid NLP pipelines which have been identified using the linguistic framework among identified pipelines by the domain expert. The precision measures the proportion of valid pipelines identified using the linguistic framework within the total number of identified pipelines. We evaluate the performance of the linguistic framework in identifying valid pipelines associated to the low and high level proposed applications as shown in Fig. 13 and Fig. 14.

The precision and recall metrics indicate that LingOnto is efficient in identifying valid NLP pipelines for high and low processing levels. Indeed, as shown in Fig. 13, the overall means of the precision associated to the English and French languages (86.3% and 92.3%) are almost 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 to these languages (86.3% and 92.3%) are better than the overall mean of the precision associated to the Arabic language (78%). This gap is explained by the fact that the Arabic language differs morphologically, syntactically and semantically from

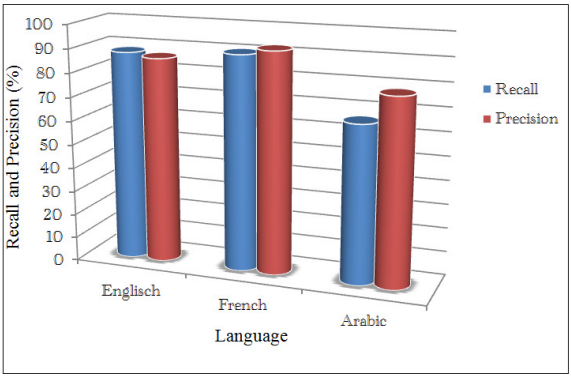

Fig. 13 Recall and Precision performances for low-level lingware applications

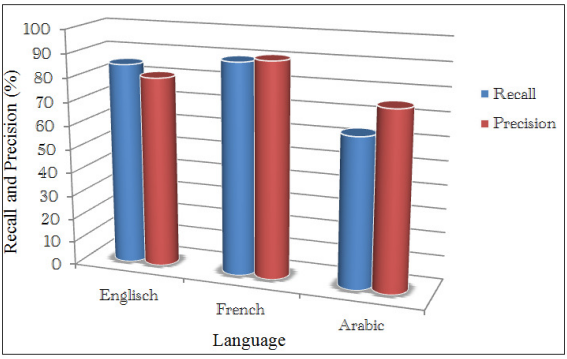

Fig. 14 Recall and Precision performance for high-level lingware applications

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 LingOnto in identifying valid pipelines for high-level Arabic applications as shown in Fig. 14. 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 Conclusion and Future Work

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. We propose an ontology based smart management of linguistic knowledge. Compared to available works, this 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, we 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, we evaluate the performance of LingOnto in identifying valid NLP pipelines for 63 proposed lingware applications. The results show that the proposed ontology is efficient in identifying valid NLP pipelines.

For future research, we plan to extend LingOnto by giving the possibility to linguistic experts adding new linguistic knowledge concepts and their associated object properties. Moreover, we suggest exploiting the NLP domain expert's feedback to improve the Not Valid identified NLP pipelines. In addition, we plan to execute the valid pipelines by discovering concrete linguistic web services that match each required linguistic processing functionality in the pipeline. Finally, we plan to allow the LingOnto ontology to be referenced by the Linked Open Vocabularies (LOV) platform.

References

[1] Gayo, J.E.L., Kontokostas, D., Auer, S.: Multilingual linked open data patterns. Semantic Web journal [under review] (2013)

[2] Ceravolo, P., Azzini, A., Angelini, M., Catarci, T., Cudré-Mauroux, P., Damiani, E., Mazak, A., Van Keulen, M., Jarrar, M., Santucci, G., *et al.*: Big data semantics. Journal on Data Semantics **7**(2), 65–85 (2018)

[3] Abderrahim, M.A., Abderrahim, M.E.A., Chikh, M.A.: Using arabic wordnet for semantic indexation in information retrieval system. arXiv preprint arXiv:1306.2499 (2013)

[4] Shinde, S.K., Bhojane, V., Mahajan, P.: Nlp based object oriented analysis and design from requirement specification. International Journal of Computer Applications **47**(21) (2012)

[5] Coletta, R., Castanier, E., Valduriez, P., Frisch, C., Ngo, D., Bellahsene, Z.: Public data integration with websmatch. In: Proceedings of the First International Workshop on Open Data, pp. 5–12 (2012)

[6] Gracia, J., Vila-Suero, D., McCrae, J.P., Flati, T., Baron, C., Dojchinovski, M.: Language resources and linked data: A practical perspective. In: International Conference on Knowledge Engineering and Knowledge Management, pp. 3–17 (2014). Springer

[7] Kless, D., Milton, S., Kazmierczak, E.: Relationships and relata in ontologies and thesauri: Differences and similarities. Applied Ontology **7**(4), 401–428 (2012)

[8] Loos, E.E.: Glossary of Linguistic Terms. SIL International, ??? (2004)

[9] Chiarcos, C., Hellmann, S.: Onlit: An ontology for linguistic terminology. In: Language, Data, and Knowledge: First International Conference, LDK 2017, Galway, Ireland, June 19-20, 2017, Proceedings, vol. 10318, p. 42 (2017). https://doi.org/10.1007/978-3-319-59888-84. Springer

[10] Ide, N., Romary, L.: A registry of standard data categories for linguistic annotation. In: 4th International Conference on Language Resources and Evaluation-LREC'04, pp. 135–138 (2004)

[11] Schuurman, I., Windhouwer, M., Ohren, O., Zeman, D.: Clarin concept registry: the new semantic registry. In: Selected Papers from the CLARIN Annual Conference 2015, October 14–16, 2015, Wroclaw, Poland, pp. 62–70 (2016). Citeseer

[12] Fellbaum, C.: Wordnet: An electronic lexical database cambridge. MA: MIT Press (1998)

[13] Jarrar, M.: The arabic ontology–an arabic wordnet with ontologically clean content. Applied Ontology (Preprint), 1–26 (2019)

[14] Farrar, S., Langendoen, D.: An OWL-DL implementation of GOLD: An ontology for the Semantic Web. Linguistic Modeling of Information and Markup Languages: Contributions to Language Technology. AW Witt and D. Metzing. Dordrecht, Springer (2010)

[15] Chiarcos, C., Sukhareva, M.: OLiA–ontologies of linguistic annotation. Semantic Web 6 (4): 379–386 (2015)

[16] de Cea, G.A., de Mon, I., Gomez-Perez, A., Pareja-Lora, A.: 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 2004., vol. 2, pp. 124–128 (2004). IEEE

[17] Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with vowl. Semantic Web **7**(4), 399–419 (2016)

[18] Liebig, T., Noppens, O.: Ontotrack: A semantic approach for ontology authoring. Web Semantics: Science, Services and Agents on the World Wide Web **3**(2-3), 116–131 (2005)

[19] Alani, H.: Tgviztab: An ontology visualisation extension for protégé. Proc. of the 2nd Workshop on Visualizing Information in Knowledge Engineering (VIKE 03) (2003)

[20] Hussain, A., Latif, K., Rextin, A.T., Hayat, A., Alam, M.: Scalable visualization of semantic nets using power-law graphs. Applied Mathematics & Information Sciences **8**(1), 355 (2014)

[21] Catenazzi, N., Sommaruga, L., Mazza, R.: User-friendly ontology editing and visualization tools: the owleasyviz approach. In: 2009 13th International Conference Information Visualisation, pp. 283–288 (2009)

[22] Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? International journal of human-computer studies **43**(5-6), 907–928 (1995)

[23] Mohanan, M., Samuel, P.: Open nlp based refinement of software requirements. International Journal of Computer Information Systems and Industrial Management Applications **8**, 293–300 (2016)

[24] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)

[25] Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In: LREC, vol. 6, pp. 48–55 (2006)

[26] Konchady, M.: Building Search Applications: Lucene, LingPipe, and Gate. Lulu. com, ??? (2008)

[27] Ishida, T.: Language grid: An infrastructure for intercultural collaboration. In: International Symposium on Applications and the Internet (SAINT'06), p. 5 (2006). IEEE

[28] Fairen-Jimenez, D., Moggach, S., Wharmby, M., Wright, P., Parsons, S., Duren, T.: Opening the gate: framework flexibility in zif-8 explored by

experiments and simulations. Journal of the American Chemical Society **133**(23), 8900–8902 (2011)

[29] Hayashi, Y., Narawa, C.: Classifying standard linguistic processing functionalities based on fundamental data operation types. In: LREC, pp. 1169–1173 (2012)

[30] Haddar, K., Hamadou, A.B.: An ellipsis resolution system for the arabic language. Int. J. Comput. Process. Orient. Lang. **22**(4), 359–380 (2009). https://doi.org/10.1142/S1793840609002159

[31] Ziad, H., McCrae, J.P., Buitelaar, P.: Teanga: a linked data based platform for natural language processing. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)

[32] Su, L.T.: The relevance of recall and precision in user evaluation. Journal of the American Society for Information Science **45**(3), 207–217 (1994)

[33] Achich, N., Algergawy, A., Bouaziz, B., König-Ries, B.: Bioontovis: An ontology visualization tool

[34] Antoniou, G., Van Harmelen, F.: Web ontology language: Owl. In: Handbook on Ontologies, pp. 67–92. Springer, ??? (2004)

[35] Baklouti, N., Bouaziz, S., Gargouri, B., Aloulou, C., Jmael, M.: 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 (2010)

[36] Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. Journal of usability studies **4**(3), 114–123 (2009)

[37] Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., Gorrell, G.: Gate teamware: a web-based, collaborative text annotation framework. Language Resources and Evaluation **47**(4), 1007–1029 (2013)

[38] Brooke, J.: Sus: a retrospective. Journal of usability studies **8**(2), 29–40 (2013)

[39] Costea, E.-A., et al.: Machine learning-based natural language processing algorithms and electronic health records data. Linguistic and Philosophical Investigations (19), 93–99 (2020)

[40] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A.: Interfacing wordnet with dolce: towards ontowordnet. Ontology and the Lexicon: A Natural Language Processing Perspective, 36–52 (2010)

[41] Gangemi, A., Guarino, N., Oltramari, A., Borgo, S.: Cleaning-up wordnet's top-level. In: Proceedings of the 1st International WordNet Conference, pp. 21–25 (2002)

[42] Hanke, H., Knees, D.: A phase-field damage model based on evolving microstructure. Asymptotic Analysis **101**, 149–180 (2017)

[43] Henrich, V., Hinrichs, E.: Gernedit: a graphical tool for germanet development. In: Proceedings of the ACL 2010 System Demonstrations, pp. 19–24 (2010)

[44] Holten, D., Van Wijk, J.J.: Force-directed edge bundling for graph visualization. In: Computer Graphics Forum, vol. 28, pp. 983–990 (2009). Wiley Online Library

[45] Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., DiPersio, D., Shi, C., Suderman, K., Verhagen, M., Wang, D., Wright, J.: The language application grid. In: International Workshop on Worldwide Language Service Infrastructure, pp. 51–70 (2015). Springer

[46] Klingström, T., Hernández-de-Diego, R., Collard, T., Bongcam-Rudloff, E.: Galaksio, a user friendly workflow-centric front end for galaxy. EMBnet. journal **23**, 897 (2017)

[47] Krivov, S., Williams, R., Villa, F.: Growl: A tool for visualization and editing of owl ontologies. Journal of Web Semantics **5**(2), 54–57 (2007)

[48] Lanzenberger, M., Sampson, J., Rester, M.: Visualization in ontology tools. In: 2009 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 705–711 (2009). https://doi.org/10.1109/CISIS.2009.178

[49] Lefever, E.: A hybrid approach to domain-independent taxonomy learning. Applied Ontology **11**(3), 255–278 (2016)

[50] Lohmann, S., Negru, S., Haag, F., Ertl, T.: Vowl 2: User-oriented visualization of ontologies. In: International Conference on Knowledge Engineering and Knowledge Management, pp. 266–281 (2014). Springer

[51] Meltzer, P.S., Kallioniemi, A., Trent, J.M.: Chromosome alterations in human solid tumors. In: Vogelstein, B., Kinzler, K.W. (eds.) The Genetic Basis of Human Cancer, pp. 93–113. McGraw-Hill, New York (2002)

[52] Murray, P.R., Rosenthal, K.S., Kobayashi, G.S., Pfaller, M.A.: Medical Microbiology, 4th edn. Mosby, St. Louis (2002)

[53] Neji, M., Ghorbel, F., Gargouri, B.: Visualizing a linguistic ontology with ling-graph. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawinski, B. (eds.) Computational Collective Intelligence - 11th International Conference, ICCCI 2019, Hendaye, France, September 4-6, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11683, pp. 41–52. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-28377-3_4. https://doi.org/10.1007/978-3-030-28377-3_4

[54] Peroni, S., Motta, E., d'Aquin, M.: 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 (2008)

[55] Schalley, A.C.: Ontologies and ontological methods in linguistics. Language and Linguistics Compass **13**(11), 12356 (2019)

[56] Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE Symposium on Visual Languages, pp. 336–343 (1996). IEEE

[57] Singh, G., Prabhakar, T., Chatterjee, J., Patil, V., Ninomiya, S., *et al.*: Ontoviz: visualizing ontologies and thesauri using layout algorithms. In: the Fifth International Conference of the Asian Federation for Information Technology in Agriculture (AFITA 2006) (2006)

[58] SWRL, A.: Semantic web rule language combining owl and ruleml. W3C Member Submission (May 21, 2004), http://www. w3. org/Submission/SWRL/(last visited March 2011) (2004)

[59] Wilcock, G.: Unstructured information management architecture (uima). In: 2nd UIMA@ GSCL Workshop (2009)

[60] Wilson, E.: Active vibration analysis of thin-walled beams. PhD thesis, University of Virginia (1991)

[61] Zhou, M., Duan, N., Liu, S., Shum, H.-Y.: Progress in neural nlp: modeling, learning, and reasoning. Engineering **6**(3), 275–290 (2020)