An Ontology based Smart Management of Linguistic Knowledge

Mariem Neji[1], Fatma Ghorbel[1,2], Bilel Gargouri[1], Nada Mimouni[2], 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: Mariem Neji , `mariem.neji@yahoo.fr`

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. In order to deal with this, a "smart" linguistic knowledge management may help the users to understand the meaning, scope and especially the use of related techniques and algorithms. In this paper, *(1)* we propose a semantic processing of linguistic knowledge based on a multilingual linguistic domain ontology, called LingOnto. Compared to related work, LingOnto 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 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)* We also propose an assisted user-friendly ontology visualization tool called LingGraph. It facilitates the interaction with LingOnto. It offers an easy to use interface for users not familiar with ontologies. It is based on a SPARQL pattern-based approach to allow a smart search interaction functionality to visualise only the ontological view corresponding to the user's needs and preferences. In order to evaluate LingOnto, we apply it to a framework of identifying valid natural language processing pipelines. Finally, we give the results of the carried-out experiments.

I INTRODUCTION

The importance of linguistic knowledge is increasing in many application areas, such as multilingual big data Gayo et al. [2013], Ceravolo et al. [2018], information retrieval Abderrahim et al. [2013], question-answering and NLP-based applications Shinde et al. [2012], data integration Coletta et al. [2012], multilingual web Gracia et al. [2014], among others.

However, handling linguistic knowledge to propose such lingware applications is a crucial issue. In order to deal with this, a smart linguistic knowledge management may help the users 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 the 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 Kless et al. [2012]. Besides, they do not support multilingualism. Ontologies are more useful as they provide more precise and semantically richer results Coletta et al. [2012]. However, most of the proposed ontologies only represent the linguistic data (e.g., Lexical unit 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 users in understanding the linguistic knowledge and proposing lingware applications. Besides, they are hard to be used by users less or not familiar with ontologies as they do not offer an ontology visualisation 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, (1) we propose a multilingual ontology called LingOnto, that covers the different aspects of the NLP domain. It aims at making a wide range of linguistic data, linguistic processing functionalities and linguistic processing features easily accessible to the users. Moreover, LingOnto enables reasoning, via a SWRL[1]-based reasoning engine, about the aforementioned knowledge in order to guide the users 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. It is designed to be used by users, who are not necessarily ontology experts. (2) To overcome this issue, we propose a user-friendly ontology visualisation tool called LingGraph. It offers an understandable visualisation 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 visualises the 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. As a use case, we have applied it to a framework of identifying valid NLP pipelines.

Our paper is organised 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 visualisation tool LingGraph. The evaluations of the performance of LingOnto will be presented in Section 5. Finally, Section 6 draws conclusions and future research directions.

II   RELATED WORK

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

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 Loos [2004] represents information based on glossaries and bibliographies proposed to support the linguistic research. This glossary supports only French

---

[1]Semantic Web Rule Language

and English linguistic terms. Moreover, it only gives 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). Finally, 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].

In Fellbaum [1998], the authors propose WordNet, which is a large lexical database that consists of a set of synsets (i.e., sets of synonyms). These latter are related with semantic relations like synonym, hyponymy and meronymy. However, these latter are not used in a consistent way as they present redundancy Fellbaum [1998]. Moreover, WordNet provides a poor classification of the types of numbers (i.e., Real, Rational and Natural, and Integer numbers are all subsumed by Number, while they subsume each other) Jarrar [2019].

The ISOcat data category registry Ide and Romary [2004] only defines 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] has taken over the work of the ISOcat data category registry. Still, it still provides very limited structural and relational information Schuurman et al. [2016].

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 linguistic data in the NLP domain. Moreover, they only focus 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 Declerck et al. [2010], the authors propose the Lemon ontology, which represents the lexical data from a Semantic Web perspective. It emerges from a combination, review and extension of prior models such as LingInfo Buitelaar et al. [2006], LexOnto Cimiano et al. [2007] and SKOS Miles and Bechhofer [2009]. Its successor, OntoLex, is the result of opening lemon to a wider community under the umbrella of the W3C Ontology-Lexica community group, in order to extend it and make it more modular. The OntoLex develops specifications for a lexicon-ontology model that can be used to provide rich linguistic grounding for domain ontologies. Rich linguistic grounding includes the representation of morphological, syntactic properties of lexical entries as well as the syntax-semantics interface (i.e., the meaning of these lexical entries with respect to the ontology in question). However, both lemon and OntoLex only focus on representing linguistic data aspect and neglect the processing one. Moreover, they do not propose a reasoning mechanism.

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 in isolation. Moreover, this ontology does not aim at captur-

ing the semantics of terms. It mainly classifies morphological notations, such as expressions, grammar, and meta-concepts Jarrar [2019]. 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 modelling annotation schemes and their linking with reference categories. Conceptually, the OLiA ontology is closely related to the OntoTag ontologies[2] ontologies proposed by de Cea et al. [2004]. One important difference is that the OntoTag ontologies are only considering the languages of the Iberian peninsula (in particular Spanish).

In Pearsall [2016], the authors propose the Oxford Global Languages Ontology (OGL), which is essentially developed to model and integrate multilingual linguistic data from Oxford Dictionaries Edwards [2010]. It includes elements to account for a range of information found in dictionaries, from inflected forms to semantic relations, pragmatic features and etymological data. This ontology allows linkage with lemon content and ontologies for linguistic descriptions. The emphasis of the approach is set on representing grammatical information with cross-linguistic validity and on maintaining grammar traditions in different languages as key points. However, OGL ontology do not represent linguistic processing functionalities and features.

In Cimiano et al. [2011], the authors propose LexInfo, which is an extensive ontology of types, values and properties partially derived from ISOcat. Currently, the elements of this ontology capture information from the morphosyntactic, syntactic, syntactic–semantic, semantic and pragmatic levels of linguistic description. However, LexInfo covers only the linguistic data and do not offer any reasoning mechanism.

In Jarrar [2019], 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 underlying structure. It currently consists of about 1,000 well-investigated concepts in addition to 11,000 concepts that are partially validated. However, this ontology does not support multilingualism, being restructed to the Arabic language.

We note that all the above-mentioned ontologies only focus on representing linguistic data and neglect the related processing issues. 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 Visualisation

In the literature, various ontology visualisation tools have been proposed. However, most of them are designed to be used by ontology experts and they overlook the importance of the usability and understandability requirements. According to Lohmann et al. [2016], the generated visualisations "are hard to read for casual users". For instance, GrOWL and SOVA[3] are intended to offer an understandable visualisation by defining notations using different symbols, colours, and node shapes for each ontology key-element. However, the proposed notations contain many abbreviations and symbols from the Description Logic domain. As a consequence, the gener-

---

[2]http://oa.upm.es/13827/
[3]http://protegewiki.stanford.edu/wiki/SOVA

ated visualisations are not suitable for non-ontology expert users. OWLViz[4], OntoTrack Liebig and Noppens [2005], KC-Viz and OntoViz show only specific element(s) of the ontology. For instance, the OWLViz and KC-Viz only visualise the class hierarchy of the 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] which provide visualisations 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 colour. This issue has a bad impact on the understandability of the generated visualisation.

Only very few visualisation 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 tools are either not available for downloading or use some Semantic Web concepts which have a negative impact on the understandability of the generated visualisation especially for the non ontology expert users.

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 keywords that the users are looking for. However, they do not offer advanced search by extracting a combination of components taking into account the users' need.

III 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[5]. 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 and assisting in the process of proposing lingware applications (e.g., it helps the users to avoid incoherency errors by assisting them selecting only compatible linguistic processing functionalities.).

3.1 Representing Linguistic Knowledge

We have relied 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, we define the top-level concepts of our ontology which are linguistic data, linguistic processing functionalities and linguistic processing features. The latter will be discussed more extensively 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 (e.g., noun phrase, verb phrase and prepositional phrase), morphosyntactic (e.g., number and gender), termi-

---

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

nological (e.g., processes, properties and functions) and lexical categories (e.g., Nouns, verbs, adjectives and adverbs).

- It supports various languages. For instance, it provides a 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. Figure. 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 lexical units).
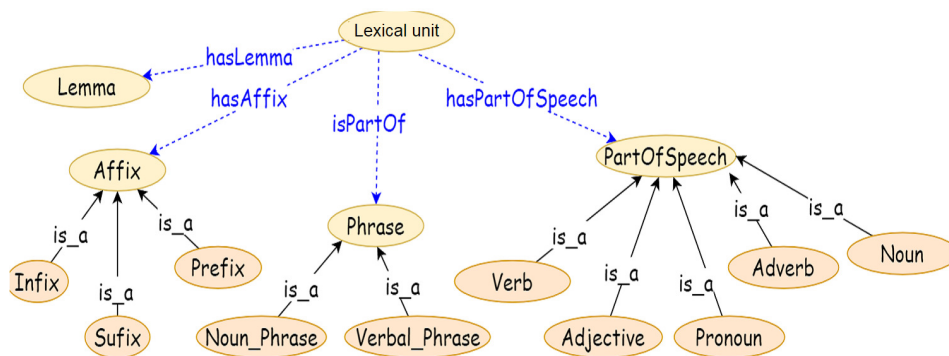

Figure 1: The classification of some Arabic linguistic data

3.1.2 *Linguistic Processing Functionalities Classification*

Referring to well-known NLP toolkits such as Apache OpenNLP Mohanan and Samuel [2016], StandfordCoreNlP 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], 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 Hayashi and Narawa [2012], 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 Figure. 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 Figure. 2, the (*"Tokenization"*) class is in relation with the (*"Sentence"*) class through *"Has_Input"* object property. It is also in relation with the (*"Lexical unit"*) class through *"Has_Output"* object property.
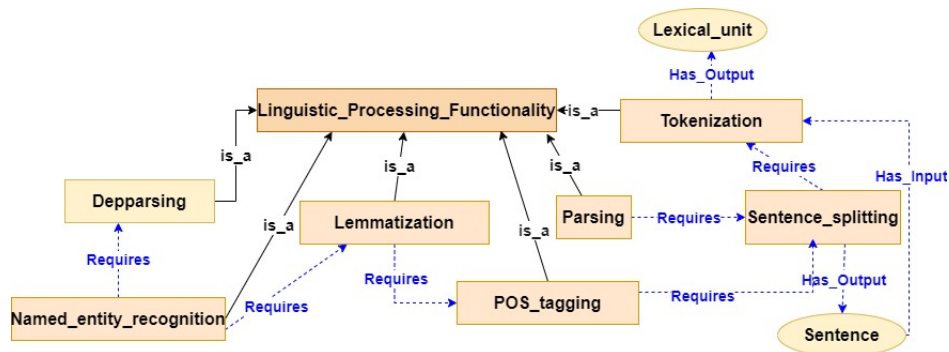

Figure 2: The classification of some Arabic linguistic processing functionalities

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 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. 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.
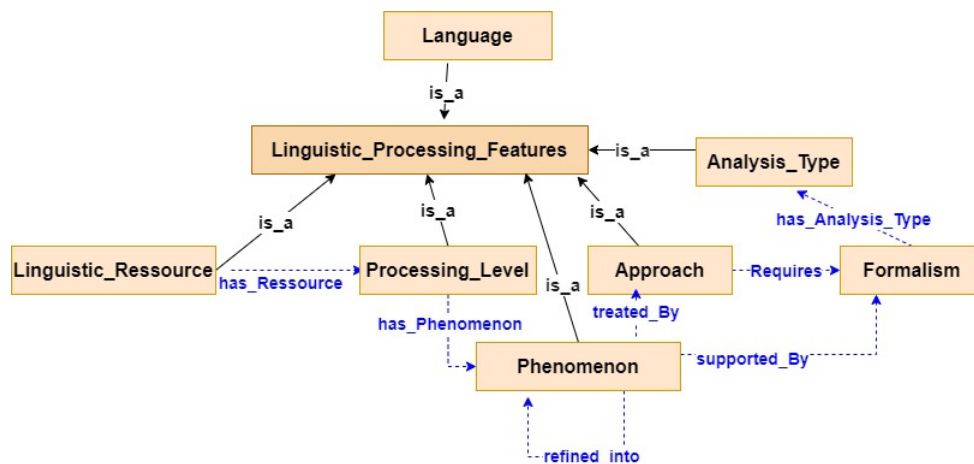

Figure 3: The classification of some linguistic processing features

3.2 Reasoning about Linguistic Knowledge

LingOnto proposes a set of SWRL rules to reason about linguistic knowledge, infer new data and assist the users in understanding the NLP domain. Compared to axioms, SWRL rules offer a predefined list of built_ins that facilitate the expression of rules such as:

- For comparison: swrlb :notEqual, swrlb :lessThan and swrlb :greaterThan.
- For strings: swrlb :stringEqualIgnoreCase and swrlb :contains.
- For lists: swrlb :member, swrlb :length and swrlb :empty.
- For dates and time: swrlb :date and swrlb :time.

SWRL rules are supported by several inference engines, such as Jess[6], Pellet Sirin et al. [2007] and Fact++ Tsarkov and Horrocks [2006].

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.
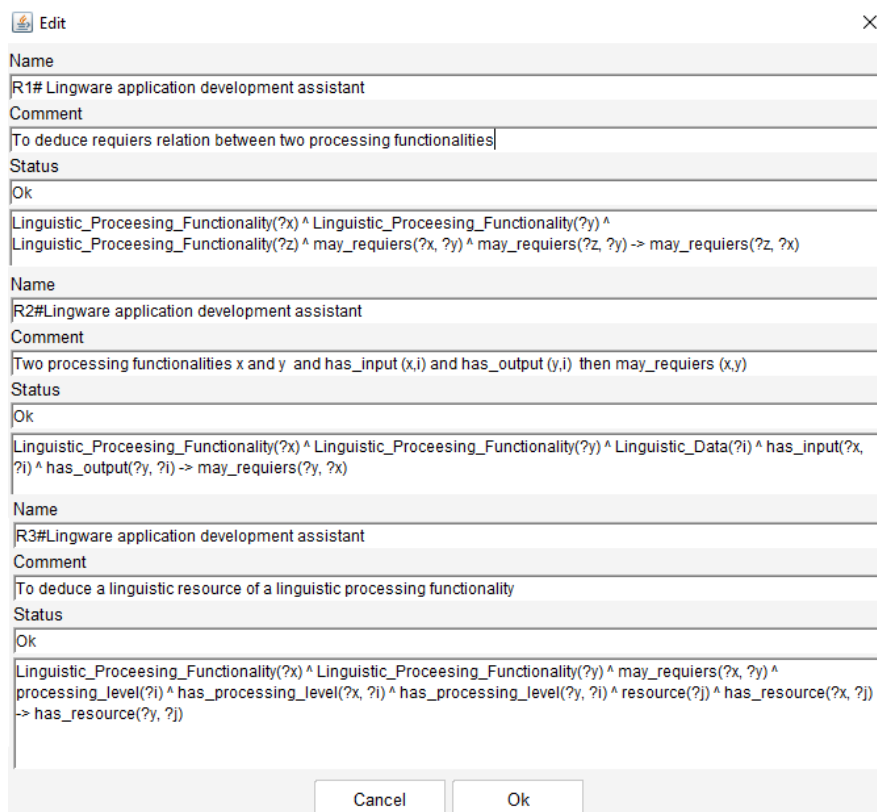
---

[6]http://herzberg. ca.sandia.gov/jess/.

3.2.1 SWRL Rules for Lingware Applications Development Assistant

LingOnto proposes a set of SWRL rules that assist the users in selecting compatible linguistic processing functionalities in order to identify valid NLP pipelines. Figure. 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 a "requires" relation between the processing functionalities "z" and "y" is inferred. This rule means that processing functionalities can be enchained in an NLP pipeline only if each one requires the other.
- 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 a "requires" relation between the processing functionalities "x" and "y" is inferred. This rule means that if a linguistic processing functionality require, as input, the results of other processing functionality; then, these latter can be enchained in an NLP pipeline.
- 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 a "use" relation between the processing functionality "y" and the linguistic resource "j" is inferred. This rule means that two enchained processing functionalities that belong to the same linguistic level should use the same linguistic resource.

Figure 4: Example of SWRL rules for lingware applications development assistant

3.2.2 SWRL Rules for NLP Domain Understanding Assistant

LingOnto proposes a set of SWRL rules to assist the users in understanding the meaning of different linguistic knowledge. Figure. 5 shows some examples.
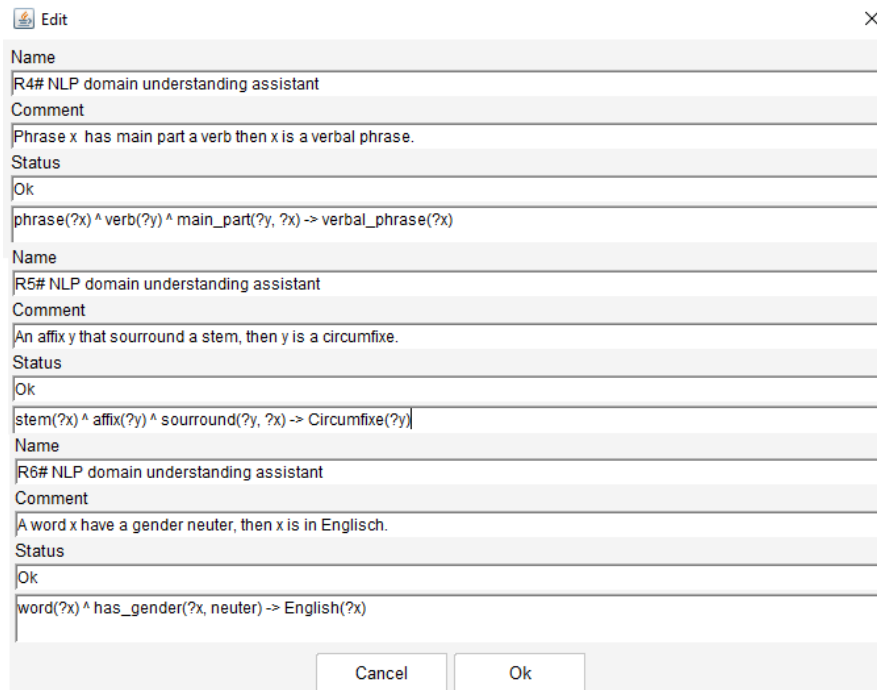
Figure 5: Example of SWRL rules for NLP domain understanding assistant.

- Rule R4 identifies if a phrase "x" has a main part a verb "y"; then the phrase "x" is a verb phrase. This rule means that if a phrase contains both the verb and either a direct or indirect object then it represents a verb phrase.
- Rule R5 identifies if an affix "y" surrounds a stem "y"; then the stem "y" is a circumfix. This rule means that if an affix has two parts were one placed at the start of a word, and the other at the end then this affix represents a circumfix.
- Rule R6 identifies if a lexical unit "x" has a gender neuter; then the lexical unit "x" is in English. Since we work only with three languages which are French, English and Arabic, if the gender of a word is neuter then this latter can be only written in English.

IV   LINGGRAPH: ONTOLOGY VISUALIZATION TOOL OF LINGONTO

The LingOnto domain ontology is designed to be used by users, who are not necessarily 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. It is freely available online[7]. 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 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.

---

[7]https://github.com/mariemNeji/Ling-Graph

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.

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 pre-defined 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 ask 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:
- 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.

Figure. 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 users 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
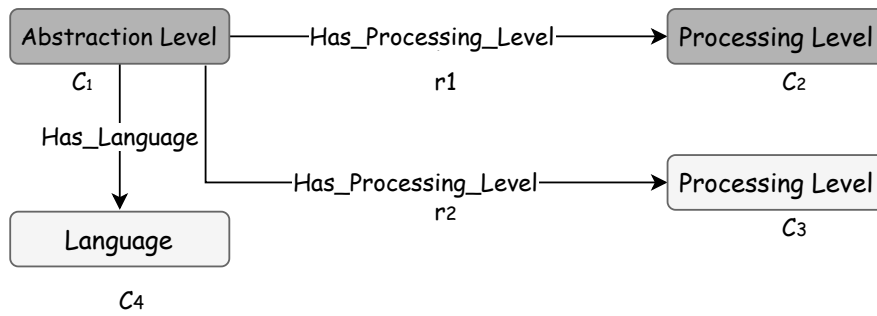
Figure 6: An Example of a pattern

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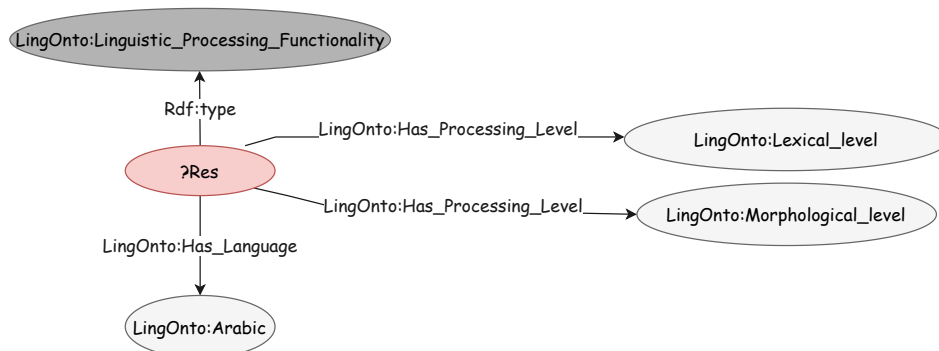

Figure 7: Query graph resulting from the instantiation of the pattern of Figure. 6

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


```
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
                }
```

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

V   EXPERIMENTATION

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

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 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. The LingOnto is applied to a framework of identifying valid NLP pipelines. It targets novice users in the lingware engineering area.

As shown in Figure. 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.

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, Figure. 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.

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 Figure. 11, the *("Diacritization")* functionality can be added to the pipeline only after *("Pos_tagging")* and *("NER")* functionalities.

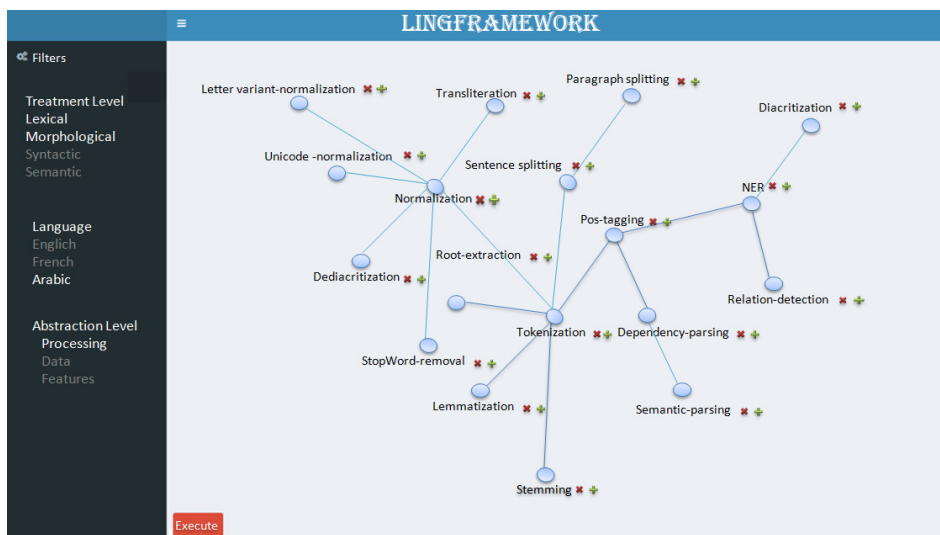The final NLP pipeline is shown in Figure. 12.

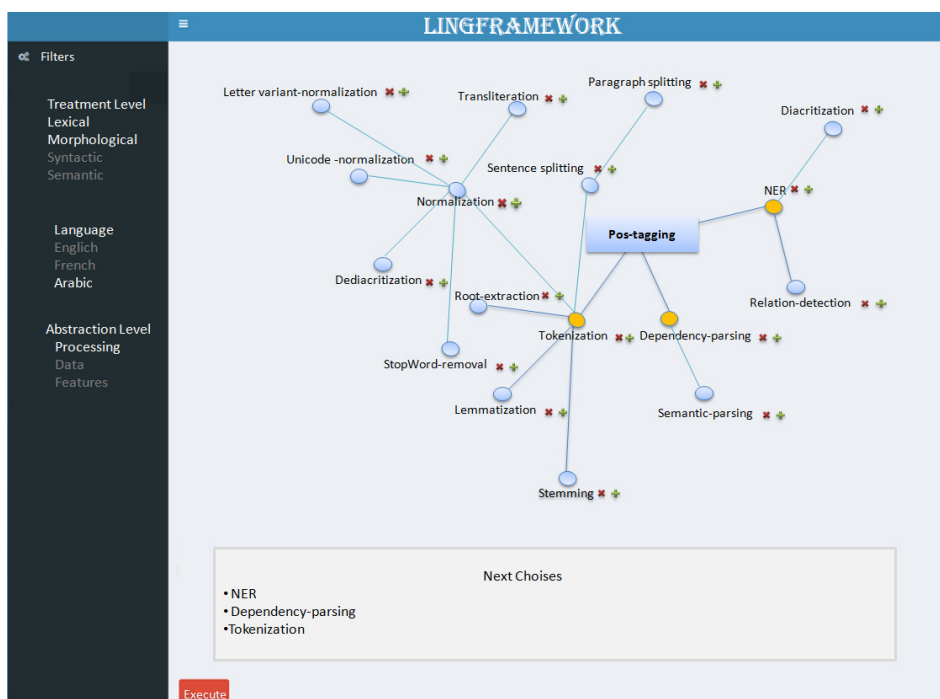Figure 9: Ontological view generation screenshot

Figure 10: NLP pipeline construction screenshot

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:

- **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, English and Arabic). Table 2 shows some examples.

- **Step 2**: we recruit three linguistic experts. The first one is a member of the Arabic Natural Language Processing Research Group (ANLP-RG) of MIRACL laboratory (Tunisia,

Figure 11: Alternative NLP pipeline proposition screenshot

Table 2: Examples of proposed lingware applications

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

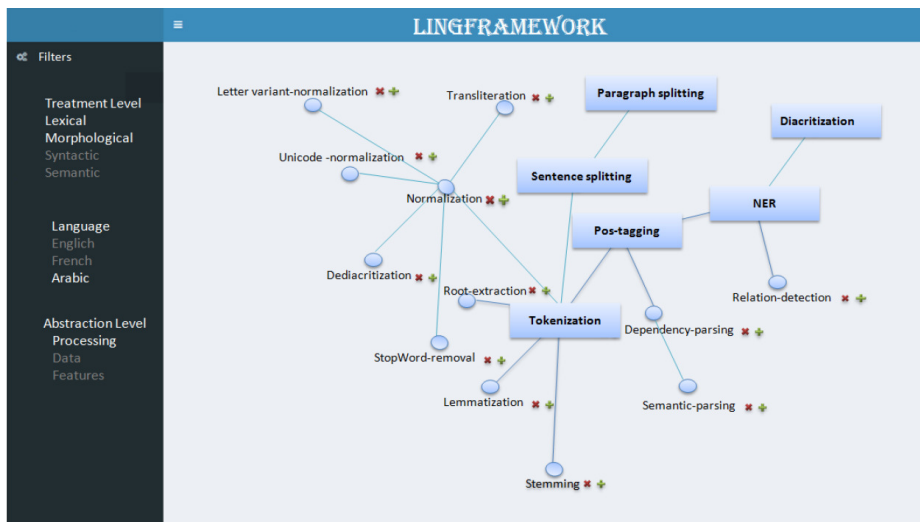Sfax). The second is a member of the CEDRIC laboratory (France, Paris). The last expert

Figure 12: The final NLP pipeline screenshot

is a member of the Formal linguistics laboratory (France, Paris). We ask each expert to provide, manually, 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 Su [1994] to evaluate the performance of LingOnto. The recall measures the proportion of valid NLP pipelines which are 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 the 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 Figure. 13 and Figure. 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 Figure. 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 have a lexical similarity (similarity in both form and meaning). Indeed, they have the same alphabet. They sometimes use similar grammatical structures and have several lexical units 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 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 Figure. 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 lexical units 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 the parsing process.

VI CONCLUSION AND FUTURE WORK

This paper addresses the issue of assisting the users 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 visual-
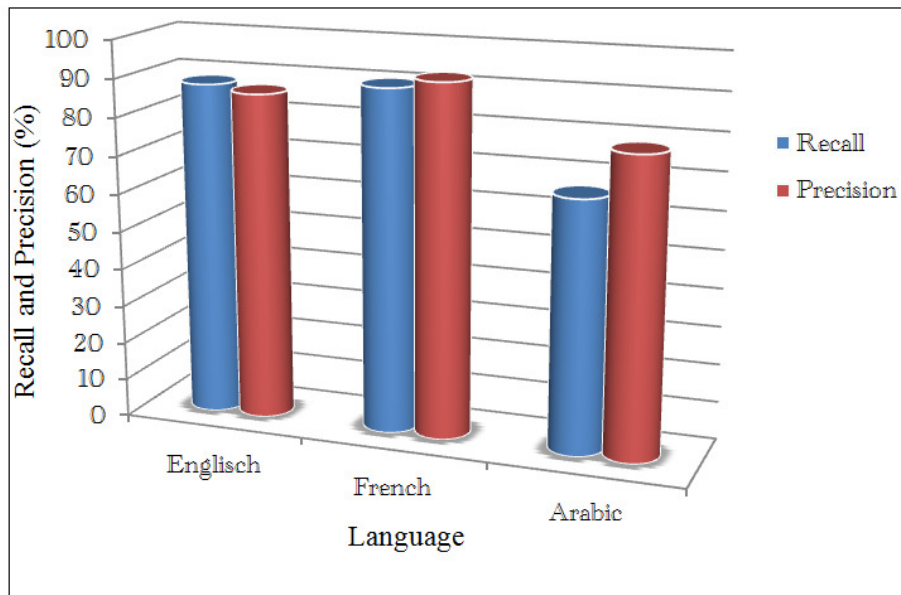
Figure 13: Recall and Precision performances for low-level lingware applications

izes an excerpt ontological view from LingOnto containing only the 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 propose an ontology maintenance collaborative tool that allows linguistic experts, independently of their native language, to manipulate LingOnto (i.e., adding and updating concepts). This tool will handle the imperfection of the proposed knowledge and the conflict between the expert's opinions. Besides, 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 re-
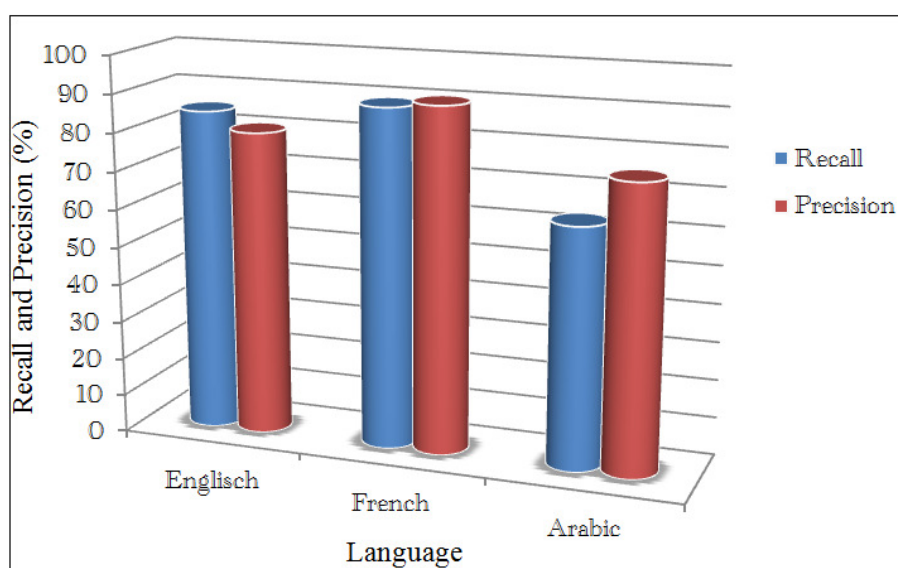

Figure 14: Recall and Precision performance for high-level lingware applications

quired 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

Mohammed Alaeddine Abderrahim, Mohammed El Amine Abderrahim, and Mohammed Amine Chikh. Using arabic wordnet for semantic indexation in information retrieval system. *arXiv preprint arXiv:1306.2499*, 2013.

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

Jordi Atserias, Bernardino Casas, Elisabet Comelles, Maritxell González, Lluís Padró, and Muntsa Padró. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *LREC*, volume 6, pages 48–55, 2006.

Paul Buitelaar, Thierry Declerck, Anette Frank, Stefania Racioppa, Malte Kiesel, Michael Sintek, Ralf Engel, Massimo Romanelli, Daniel Sonntag, Berenike Loos, et al. Linginfo: Design and applications of a model for the integration of linguistic information in ontologies. In *Proceedings of the OntoLex Workshop at LREC*, 2006.

Nadia Catenazzi, Lorenzo Sommaruga, and Riccardo Mazza. User-friendly ontology editing and visualization tools: the owleasyviz approach. In *2009 13th International Conference Information Visualisation*, pages 283–288, 2009.

Paolo Ceravolo, Antonia Azzini, Marco Angelini, Tiziana Catarci, Philippe Cudré-Mauroux, Ernesto Damiani, Alexandra Mazak, Maurice Van Keulen, Mustafa Jarrar, Giuseppe Santucci, et al. Big data semantics. *Journal on Data Semantics*, 7(2):65–85, 2018.

C Chiarcos and M Sukhareva. Olia–ontologies of linguistic annotation. semantic web 6 (4): 379–386, 2015.

Christian Chiarcos and Sebastian Hellmann. Onlit: An ontology for linguistic terminology. In *Language, Data, and Knowledge: First International Conference, LDK 2017, Galway, Ireland, June 19-20, 2017, Proceedings*, volume 10318, page 42. Springer, 2017. doi: 10.1007/978-3-319-59888-84.

Philipp Cimiano, Peter Haase, Matthias Herold, Matthias Mantel, and Paul Buitelaar. Lexonto: A model for ontology lexicons for ontology-based nlp. In *Proceedings of the OntoLex07 Workshop held in conjunction with ISWC 07*. Citeseer, 2007.

Philipp Cimiano, Paul Buitelaar, John McCrae, and Michael Sintek. Lexinfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics*, 9(1):29–51, 2011.

Remi Coletta, Emmanuel Castanier, Patrick Valduriez, Christian Frisch, DuyHoa Ngo, and Zohra Bellahsene. Public data integration with websmatch. In *Proceedings of the First International Workshop on Open Data*, pages 5–12, 2012.

Guadalupe Aguado de Cea, IA de Mon, Asuncion Gomez-Perez, and Antonio Pareja-Lora. 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.*, volume 2, pages 124–128. IEEE, 2004.

Thierry Declerck, Paul Buitelaar, Tobias Wunner, J McCrae, Elena Montiel-Ponsoda, and G Aguado de Cea. Lemon: An ontology-lexicon model for the multilingual semantic web. 2010.

Helen Edwards. Oxford dictionaries online. *Refer*, 26(2):A20, 2010.

D Fairen-Jimenez, SA Moggach, MT Wharmby, PA Wright, S Parsons, and T Duren. Opening the gate: framework flexibility in zif-8 explored by experiments and simulations. *Journal of the American Chemical Society*, 133 (23):8900–8902, 2011.

S Farrar and DT Langendoen. 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, 2010.

Christiane Fellbaum. Wordnet: An electronic lexical database cambridge. *MA: MIT Press*, 1998.

Jose Emilio Labra Gayo, Dimitris Kontokostas, and Sören Auer. Multilingual linked open data patterns. *Semantic Web journal [under review]*, 2013.

Jorge Gracia, Daniel Vila-Suero, John P McCrae, Tiziano Flati, Ciro Baron, and Milan Dojchinovski. Language resources and linked data: A practical perspective. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 3–17. Springer, 2014.

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

Kais Haddar and Abdelmajid Ben Hamadou. An ellipsis resolution system for the arabic language. *Int. J. Comput. Process. Orient. Lang.*, 22(4):359–380, 2009. doi: 10.1142/S1793840609002159. URL https://doi.org/10.1142/S1793840609002159.

Yoshihiko Hayashi and Chiharu Narawa. Classifying standard linguistic processing functionalities based on fundamental data operation types. In *LREC*, pages 1169–1173, 2012.

Ajaz Hussain, Khalid Latif, Aimal Tariq Rextin, Amir Hayat, and Masoon Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355, 2014.

Nancy Ide and Laurent Romary. A registry of standard data categories for linguistic annotation. In *4th International Conference on Language Resources and Evaluation-LREC'04*, pages 135–138, 2004.

Toru Ishida. Language grid: An infrastructure for intercultural collaboration. In *International Symposium on Applications and the Internet (SAINT'06)*, pages 5–pp. IEEE, 2006.

Mustafa Jarrar. The arabic ontology–an arabic wordnet with ontologically clean content. *Applied Ontology*, (Preprint):1–26, 2019.

Daniel Kless, Simon Milton, and Edmund Kazmierczak. Relationships and relata in ontologies and thesauri: Differences and similarities. *Applied Ontology*, 7(4):401–428, 2012.

Manu Konchady. *Building Search Applications: Lucene, LingPipe, and Gate*. Lulu. com, 2008.

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

Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with vowl. *Semantic Web*, 7(4):399–419, 2016.

Eugene Emil Loos. *Glossary of linguistic terms*. SIL International, 2004.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

Alistair Miles and Sean Bechhofer. Skos simple knowledge organization system reference. *W3C recommendation*, 2009.

Murali Mohanan and Philip Samuel. Open nlp based refinement of software requirements. *International Journal of Computer Information Systems and Industrial Management Applications*, 8:293–300, 2016.

Judy Pearsall. Oxford global languages. *Dictionaries: Journal of the Dictionary Society of North America*, 37(1): 165–177, 2016.

Ineke Schuurman, Menzo Windhouwer, Oddrun Ohren, and Daniel Zeman. Clarin concept registry: the new semantic registry. In *Selected Papers from the CLARIN Annual Conference 2015, October 14–16, 2015, Wroclaw, Poland*, number 123, pages 62–70. Citeseer, 2016.

Subhash K Shinde, Varunakshi Bhojane, and Pranita Mahajan. Nlp based object oriented analysis and design from requirement specification. *International Journal of Computer Applications*, 47(21), 2012.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.

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

Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *International joint conference on automated reasoning*, pages 292–297. Springer, 2006.

Housam Ziad, John Philip McCrae, and Paul Buitelaar. 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.