Retrieving Information from the French Lexical Network in RDF/OWL Format

Alexsandro Fonseca[1], Fatiha Sadat[1] and François Lareau[2]
[1]University of Quebec in Montreal, [2]University of Montreal
[1]201 President Kennedy, Montreal, Canada, [2]C.P. 6128 succ. Centre-Ville, Montreal, Canada
affonseca@gmail.com, sadat.fatiha@uqam.ca, francois.lareau@umontreal.ca

Abstract

In this paper, we present a Java API to retrieve the lexical information from the French Lexical Network, a lexical resource based on the Meaning-Text Theory's lexical functions, which was previously transformed to an RDF/OWL format. We present four API functions: one that returns all the lexical relations between two given vocables; one that returns all the lexical relations and the lexical functions modeling those relations for two given vocables; one that returns all the lexical relations encoded in the lexical network modeled by a specific lexical function; and one that returns the semantic perspectives for a specific lexical function. This API was used in the identification of collocations in a French corpus of 1.8 million sentences and in the semantic classification of these collocations.

Keywords: lexical network, lexical functions, Meaning-Text Theory, ontology, RDF, OWL

1. Introduction

The languages RDF/OWL have been an important tool for building interconnected resources in the web, due to their simplicity. RDF allows the construction of knowledge graphs. OWL allows the inference of logical relations among the objects represented in those graphs and the creation of classes of objects. RDF and OWL are, to date, the most successful knowledge representation languages (Hendler and van Harmelen, 2008). The set of resources in RDF/OWL format that are connected to each other through the internet is known as the Semantic Web.

Linguistic resources have been developed on top of RDF/OWL or transformed into an RDF/OWL format. As examples, we cite: WordNet (Fellbaum, 1998), DBPedia Wiktionary,[1] FrameNet (Fillmore, 1977), etc.

For a more detailed representation of linguistic information, however, the RDF/OWL languages are not sufficient. For this reason, metalinguistic ontologies were developed to represent information such as *part of speech*, *direct object*, *noun phrase*, etc. Those metalinguistic ontologies evolved into the *lexicon model for ontologies* (lemon) (McCrae et al., 2011), the most recent ISO standard for the representation of lexical information in the Semantic Web.

We have developed a metalinguistic ontology (*lexfom*) to represent Meaning Text Theory's (MTT) lexical functions. *Lexfom* uses the *lemon* model to represent information about lexical entries and lexical senses. This ontology was applied in the transformation of the French Lexical Network into an RDF/OWL format.

In this paper, we present a Java API that was developed to retrieve the lexical and combinatorial information from the French Lexical Network, which is based on lexical functions, in an RDF/OWL format.

This paper is divided as follows. In §2., we present the notions behind our API: the lexical functions and the French Lexical Network, a semantic classification of lexical functions and the metalinguistic ontologies, including the ontology that we have developed to represent MTT's lexical functions.

In §3., we present the functions that we have developed in our API to retrieve information from the French Lexical Network in RDF/OWL format. In §4., we conclude and discuss future work.

2. Related Work

2.1. The French Lexical Network

To our knowledge, the *French Lexical Network* (FLN) (Lux-Pogodalla and Polguère, 2011) is the only lexical network based on lexical functions. It has been developed as part of the *RELIEF* project[2] at *ATILF*.[3]

Unlike other lexical networks, such as WordNet (Fellbaum, 1998), the FLN does not make a taxonomic classification of words (Polguère, 2014). Moreover, the FLN contains syntagmatic relations between lexemes, usually absent from other lexical networks.

In this paper, we adopt the nomenclature used by the MTT: the term *vocable* refers to a canonical form of a word, independent of its meaning. The term *lexeme* refers to a specific acceptation of a vocable. For example, the vocable *mouse* has two different lexemes, $mouse_I$ (an animal) and $mouse_{II}$ (a computer device).

A lexical function (LF) (Mel'čuk, 1998) is a linguistic tool to represent different types of relations between lexemes.

Those relations can be paradigmatic, such as synonymy, antonymy and hyperonymy, or syntagmatic (horizontal relations in a sentence or collocation), such as intensification (e.g. *strongly condemn*) and subjective qualification (e.g. *fruitful analysis*).

LFs have the following general format: LF (*base*) = *value*. The *value* is a set of one or more lexemes. For example: Anti (*small*) = {*big*}; Hyper (*cat*) = {*feline*, *mammal*, *animal*}; Magn (*applause*) = {*thunderous*}. Simple LFs can be combined to form complex LFs: AntiMagn (*applause*) = {*scattered*}.

[1] www.dbpedia.org/page/Wiktionary

[2] www.atilf.fr/spip.php?article908

[3] Analyse et Traitement de la Langue Française: www.atilf.fr

The FLN is available for download in an XML format on ORTOLANG[4] (ATILF, 2017).
Since the information in the FLN is encoded in an XML format, and not in RDF/OWL, they cannot be immediately connected to the Semantic Web. Moreover, the information about LFs is only textual. This means that we do not have, for example, the following information:

- How complex LFs are formed from simple LFs. For example, that the LF $AntiMagn$ is composed from the LFs $Anti$ and $Magn$;
- How an LF like $Oper_1$ is related to the LFs $Real_1$ or $Func_1$ through the first actant (represented by the index *1*);
- Information about the semantic perspective of a lexical function (presented in the next section);

For this reason, we have developed a metalinguistic model called *lexfom*[5] (Fonseca et al., 2016a), which is presented in §2.4., to represent the characteristics of LFs and we have applied this model in the transformation of the FLN into an RDF/OWL format[6].

2.2. Semantic perspective for lexical functions

Jousse (2010) presents four different classifications for LFs: a semantic, a pragmatic, a combinatorial and a syntactic classification. These classifications are called "perspectives". In this paper, we are interested in the semantic perspective (SP).
The SP is comprised of ten classes: *action/event*, *causativity*, *element/set*, *equivalence*, *location*, *opposition*, *participants*, *phase/aspect*, *qualification* and *utilization form*. We added two classes to this classification, *semantically empty verb* and *support verb*.
Some of those classes have sub-classes. For example, the class *qualification* is sub-divided into *intensity* (e.g. Magn (*shave*) = {*close*}), *positive evaluation* (e.g. Bon (*contribution*) = {*valuable*}), and *negative evaluation* (e.g. $\text{AntiBon}_1\text{Involv}$ (*car*) = {*smash into* N}, where N represents a noun).
Finally, the lexical relation between lexemes modeled by a specific LF can be classified in the same way.

2.3. Metalinguistic ontologies

The languages RDF/OWL only allow the representation of simple statements, encoded as triplets. For the representation of more complex linguistic information, metalinguistic ontologies based on RDF/OWL had to be developed.
The first metalinguistic ontology based on RDF/OWL was *ISOCat* (*ISO TC37 Data Category Registry*).[7] It was proposed and developed by the Psycholinguistic Department of the Max Planck Institute.[8] Its aim is to define grammatical categories, such as transitive and intransitive verbs, part of speech, predicate, etc.
Another important metalinguistic ontology is the *Lexical Markup Framework* (LMF) (Francopoulo et al., 2006). LMF is an ISO project that started in 2005 and was first published in 2007. Its aim is to be a common standard in the development of dictionaries for the Semantic Web. It is designed to represent morphological, syntactic and semantic information.
Some other metalinguistic ontologies were developed after LMF, leading to the publishing of a new W3C standard in 2016, called *lexicon model for ontologies* (lemon).[9] Lemon is based in previous models, such as LMF, ISOCat, LexInfo,[10] etc.
Lemon's main modules are the following: *Ontology-lexicon interface* (ontolex), *Syntax and Semantics* (synsem), *Decomposition* (decomp), *Variation and Translation* (vartrans) and *Linguistic Metadata* (lime).
The *ontolex* module implements a *LexicalEntry* object, which is used to represent a canonical form of a word, and a *LexicalSense* object, which is used to represent each specific sense of a word.
In our model, which is presented in the next section, each vocable and lexeme are represented by a *ontolex LexicalEntry* and a *LexicalSense* object, respectively.

2.4. Lexical functions ontology model

The *Lexical functions ontolgoy model* (lexfom) (Fonseca et al., 2016a; Fonseca et al., 2016b) is a metalinguistic ontology of lexical functions and lexical relations.
It comprises four modules:

- *Lexical functions representation* (lfrep) represents an LF's characteristics, such as its semantic actants;
- *Lexical functions relation* (lfrel) represents a relation between lexemes, which can be paradigmatic or syntagmatic;
- *Lexical functions family* (lffam) represents a syntactic classification for LFs. For example, the LF $Oper_1$ and the complex LFs composed by $Oper_1$ belong to the same family;
- *Lexical functions semantic perspective* (lfsem) is a semantic classification of LFs, based on the work of (Jousse, 2010).

We apply our model to create an RDF/OWL version of the FLN. About 46,000 paradigmatic relations and 8,000 syntagmatic relations extracted from the FLN are represented in an RDF/OWL format using the *lexfom* model.
Figure 1 shows the RDF code, in Turtle dialect,[11] representing the French collocation *porter un vêtement* (to wear a piece of clothing) using *lexfom*'s four modules and the

[4] www.ortolang.fr/market/item/lexical-system-fr/v1
[5] https://github.com/alex-fonseca/lexfom
[6] https://github.com/alex-fonseca/rlfowl
[7] www.isocat.org
[8] www.mpi.nl
[9] www.w3.org/community/ontolex
[10] lexinfo.net
[11] www.w3.org/TR/turtle

lemon's *ontolex* module. Not only the collocation is represented, but also each vocable with all their meanings found in the FLN and the LF modeling the syntagmatic relation in the collocation ($Real_1$ (*vêtement* = {*porter*}).

```
:lex_vetement a ontolex:LexicalEntry,
           ontolex:Word;
     ontolex:canonicalForm :form_vetement;
 ontolex:sense :vetement_sense_I.2;
 ontolex:sense :vetement_sense_I.1;
 ontolex:sense :vetement_sense_II;
 ontolex:sense :vetement_sense_III.1;
 ontolex:sense :vetement_sense_III.2;
     rdfs:label "vetement"@fr .

:form_vetement a ontolex:Form;
     ontolex:writtenRep "vetement"@fr .

vetement_sense_I.2 a ontolex:LexicalSense .
vetement_sense_I.1 a ontolex:LexicalSense .
vetement_sense_II a ontolex:LexicalSense .
vetement_sense_III.1 a ontolex:LexicalSense .
vetement_sense_III.2 a ontolex:LexicalSense .


:lex_porter a ontolex:LexicalEntry, ontolex:Word;
    ontolex:canonicalForm :form_porter;
        ontolex:sense :porter_sense_I.1;
        ontolex:sense :porter_sense_IV;
    rdfs:label "porter"@fr .

:form_porter a ontolex:Form;
     ontolex:writtenRep "porter"@fr .

porter_sense_I.1 a ontolex:LexicalSense .
porter_sense_IV a ontolex:LexicalSense .


LF-Real1 rdf:type lfrep:simpleLF,
           owl:NamedIndividual ;
   lfrep:belongsToLFFamily
           lffam:LFF-synt-realV-Real1;
   lfrep:hasSyntActant
           lfrep:lfrep-const-sa-ASynt_1;
   lfrep:dimension
           lfrep:lfrep-type-syntagmaticLF;
   lfrep:semanticPerspective
    lfsem:pSem-ae-utilizationTypicalOperation.


:lfsr_11420 a lfrel:SyntagmaticLFSenseRelation;
     lfrel:hasLexicalFunction lfrep:LF-Real1;
     lfrel:hasLFKeyword
                  ontolex:vetement_sense_I.2;
     lfrel:hasLFValue ontolex:porter_sense_IV;
     lfrel:hasGovPattern
                  lfgpat:"DET ~s"^^xsd:string;
     lfrel:relationDirection lfrel:valueKeyword;
     lfrel:hasFusedElement "false"^^xsd:boolean.
```

Figure 1: RDF code representing the vocables *vêtement* and *porter*, each lexical sense of both vocables, the LF $Real_1$, and finally the syntagmatic relation between a specific lexeme of each vocable.

3. Java API

In this section, we present the Java API to retrieve information from the FLN in RDF/OWL format.

3.1. API's general vision

We implemented different functions to retrieve information from the RLF in RDF/OWL format[12], using the SPARQL query language. Our implementation uses the Apache Jena ARQ,[13] a query engine implementing SPARQL.
The main functions in our API are:

- *getLexicalRelationForVocables (String vocable1, String vocable2, int typeRelation)*: given two vocables v_1 and v_2, this function returns the lexical relations (paradigmatic or syntagmatic) present in the RLF between any sense of v_1 and v_2. It is possible to search only for syntagmatic or paradigmatic relations between v_1 and v_2, by setting the variable *typeRelation*. This function is useful, for example, for applications searching for collocations, as shown in (Fonseca et al., 2017);

- *getLexicalRelationLFForVocables (String vocable1, String vocable2, int typeRelation)*: the difference between this function and the last one is the possibility of also searching for the LF modeling the relation between any senses of the vocables;

- *getLexicalRelationsForLF (String lf, typeRelation)*: given a LF *lf*, it is possible to find all the lexical relations modeled by *lf* in the FLN. It is also possible to specify only syntagmatic or paradigmatic relations;

- *getSemanticPerspectives (String lf)*: it returns all the semantic perspectives for a specific LF. Since some LFs are complex, they can have more than one semantic perspective. This function is useful, for example, in the identification of the semantic relation connecting the lexemes in a collocation. For example, the collocation *good review* is modeled by the LF *Bon*: Bon (*review*) = {*good*}. By identifying such a collocation in a text, we can find in the FLN that it is modeled by the LF *Bon* and that this LF has a semantic perspective of "positive evaluation". This information can be useful for applications in sentiment analysis, for example.

In the next subsection we show how the second function presented above is used in the identification of collocations from a corpus.

3.2. Identification of collocations

As an example of the application of this API, we applied it in the identification of collocations, as presented in (Fonseca et al., 2017). About 1.8 million phrases from the French part of the Eurosense corpus (Delli Bovi et al., 2017) were extracted and a dependency parser was applied to them. The dependency relations found in the corpus are searched in the FLN's syntagmatic relations, using the function *getLexicalRelationLFForVocables(String vocable1, String vocable2, int typeRelation)*. The positive matches are kept as possible collocations and later manually analyzed to decide if they are true collocations.
Fourteen different dependency relations are tested. We show here examples of five of these relations:

- a_obj: argument introduced by the preposition *à* - *à fond* (*thoroughly*);

- mod: modifiers (adjectival, nominal and adverbial) - *politique véritable* (*true policy*);

- obj: object of a verb - *traiter les maladies* (*to treat diseases*);

[12] https://github.com/alex-fonseca/lexfom-api

[13] jena.apache.org/documentation/javadoc/arq/org/apa-che/jena/query/package-summary.html

- p_obj: argument introduced by a preposition (other than *à* and *de* - *sur la table* (*on the table*);

- dep_coord: links a conjunct to the previous coordinator - *dans le car* (*in the car*);

The advantage of using dependency parsing combined with the FLN is shown by the following sentence: "*Quel pouvoir sur les âmes Hussey exerce-t-elle encore?*" (What power over souls is Hussey still exerting?). In this example, there is a dependency relation (*obj*) between *pouvoir* and *exerce*. The pair (*pouvoir*, *exerce*) can be searched in the FLN and the collocation *exercer le pouvoir* will be retrieved, together with the LF modeling this collocation: Oper_1 ($pouvoir_{II}$) = $exercer_{II.1}$. By this method, such a collocation can be identified, even though the vocables *pouvoir* and *exerce* are distant in the sentence.
Table 1 shows the precision for some dependencies and the total precision for the 14 dependencies. The complete table for all 14 dependency types is presented in (Fonseca et al., 2017).

Table 1: Precision in the identification of collocations by syntactic dependency.

dependency	# candidates	# true coll.	precision
mod	20 625	14 240	0.690
obj	4 869	4 720	0.969
a_obj	300	295	0.983
dep_coord	246	13	0.053
p_obj	90	86	0.956
Total	**43 629**	**33 273**	**0.763**

The most similar work to ours in the identification of collocations is the one presented by (Garcia et al., 2017). They identify collocations from three pairs of parallel corpora: English-Spanish, English-Portuguese and Spanish-Portuguese. The main difference between their work and ours is that they only use three dependencies: adjectival modifiers (*amod*), nominal modifiers (*nmod*) and verb-object (*vobj*), which are less likely to produce errors, since the governor and the dependent are adjacent to each other. Their average precision for the three language pairs are: 91.8% for *amod*, 90.6% for *nmod* and 86.2% for *vobj*.
In general, we expected to have good precision for all types of dependencies, since each candidate is matched against the collocations represented in the ontology and the ontology is based on the FLN, which is manually constructed.
However, we had false positives due to parsing errors. The most common are the errors connected to false positive collocations formed by the verbs: *pouvoir* (can) (35.1%), *avoir* (have) (31.1%) and *être* (be) (29.5%), as explained in (Fonseca et al., 2017).
Another frequent error is connected to the conjunction *car* (because), which is homonymous with the noun *car* (*bus*). In collocations like *dans le car* (*inside the bus*), it was often mistakenly tagged as a conjunction, with the dependency *dep_coord*. This explains why candidates in this group had low precision.

3.3. Classification of collocations in semantic categories

The fourth function presented in §3.1. is used in the semantic classification of collocations. The function *getSemanticPerspectives (String lf)* is used in the identification of the SP of each LF modeling each identified collocation.
For example, the API's function used to retrieve the collocation *exercer le pouvoir* (to exert power) from the FLN, presented in the previous subsection, also retrieves the LF Oper_1, which models the syntagmatic relation between the lexemes $pouvoir_{II}$ and $exercer_{II.1}$. We then use the function *getSemanticPerspectives* ($Oper_1$), which returns the SP *supportVerb*. By this method, we can identify the semantic category of each collocation.
As presented in (Fonseca et al., 2017), the main SPs for collocations identified from the EuroSense corpus are:

- *qualification* (33.9%). Example: *très grave* (very serious) - Magn (*grave*) = {*très*}.

- *supportVerb* (24.4%). Example: *exercer le pouvoir* (to exert power) - Oper_1 (*pouvoir*) = {*exercer*}.

- *location* (17.9%). Example: *dans le pays* (in the country) - Loc_{in} (*pays*) = {*dans*}.

- *actionEvent* (9.7%). Example: *l'avion atterrit* (the plane lands) - FinFact_0 (*avion*) = {*atterrir*}.

4. Conclusion and Future Work

The FLN is unique in the sense that it is the only lexical network based on lexical functions and the only one to represent syntagmatic relations between lexemes in a graph-based architecture.
The FLN is available for download in XML. Using a metalinguistic ontology created to represent lexical functions (Fonseca et al., 2016a; Fonseca et al., 2016b), we have created an RDF/OWL version of the relations inside the FLN.
In this paper, we presented a Java API developed to retrieve information from the RDF/OWL version of the RLF. We showed two applications for this API: the identification of collocations from a textual corpus and the semantic classification of the identified collocations.
As future work, we intend to connect each sense in the FLN to the senses in DBPedia, creating a stronger connection between the FLN and the Semantic Web.

5. Acknowledgements

We thank Professor Alain Polguère, director of the project RELIEF, for kindly providing us with access to the French Lexical Network before its public release.

6. Bibliographical References

ATILF. (2017). Réseau lexical du français (rl-fr). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Fonseca, A., Sadat, F., and Lareau, F. (2016a). Lexfom: a lexical functions ontology model. In *Proceedings of the Fifth Workshop on Cognitive Aspects of the Lexicon (CogALex), COLING*, pages 145–155, Osaka, Japan.

Fonseca, A., Sadat, F., and Lareau, F. (2016b). A lexical ontology to represent lexical functions. In *Proceedings of the 2nd Workshop on Language and Ontologies (OntoLex), LREC*, pages 69–73, Portorož, Slovenia.

Fonseca, A., Sadat, F., and Lareau, F. (2017). Combining dependency parsing and a lexical network based on lexical functions for the identification of collocations. In Ruslan Mitkov, editor, *Computational and Corpus-Based Phraseology*, pages 447–461, Cham. Springer International Publishing.

Garcia, M., García-Salido, M., and Alonso-Ramos, M. (2017). Using bilingual word-embeddings for multilingual collocation extraction. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 21–30, Valencia, Spain, April. Association for Computational Linguistics.

Hendler, J. and van Harmelen, F. (2008). The semantic web: Webizing knowledge representation. In Frank van Harmelen, et al., editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 821–839. Elsevier.

Jousse, A.-L. (2010). *Modèle de structuration des relations lexicales fondé sur le formalisme des fonctions lexicales*. Ph.D. thesis. Thèse de doctorat dirigée par Sylvain Kahane et Alain Polguère, Université de Montréal et Université Paris Diderot (Paris 7).

Lux-Pogodalla, V. and Polguère, A. (2011). Construction of a French Lexical Network: Methodological Issues. In *First InternationalWorkshop on Lexical Resources, WoLeR 2011*, pages 54–61, Ljubljana, Slovenia, August.

McCrae, J., Spohr, D., and Cimiano, P. (2011). Linking lexical resources and ontologies on the semantic web with lemon. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 245–259, Berlin, Heidelberg. Springer-Verlag.

Mel'čuk, I. (1998). Collocations and lexical functions. *Phraseology. Theory, Analysis and Applications*, pages 23–53.

Polguère, A. (2014). From writing dictionaries to weaving lexical networks. *International Journal of Lexicography*, 27(4):396–418.

7. Language Resource References

Delli Bovi, C., Collados, J. C., Raganato, A., and Navigli, R. (2017). Eurosense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of 55th annual meeting of the Association for Computational Linguistics (ACL 2017)*, Vancouver, Canada.

Christiane Fellbaum, editor. (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May.

Fillmore, C. J., (1977). *Scenes-and-frames semantics*. Number 59 in Fundamental Studies in Computer Science. North Holland Publishing.

Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, Y., and Soria, C. (2006). Lexical Markup Framework (LMF. In *Proceedings of the International Conference on Language Resources and Evaluation - LREC-2006*.