Combining dependency parsing and a lexical network based on lexical functions for the identification of collocations

Alexsandro Fonseca[1], Fatiha Sadat[1], and François Lareau[2]

[1] University of Quebec in Montreal, Computer Science Department
201, President Kennedy Avenue, Montreal, QC, Canada, H2X 3Y7
[2]University of Montreal, Linguistic and Translation Department
C.P. 6128, succ. Centre-Ville, Montreal, QC, Canada, H3C 3J7
fernandes_da_fonseca.alexsandro@courrier.uqam.ca
sadat.fatiha@uqam.ca
francois.lareau@umontreal.ca

Abstract. A collocation is a type of multiword expression formed by two parts: a base and a collocate. Usually, in a collocation, the base has a denotative or literal meaning, while the collocate has a connotative meaning. Examples of collocations: *pay attention*, *easy as pie*, *strongly condemn*, *lend support*, etc. The Meaning-Text Theory created the lexical functions to, among other objectives, represent the meaning existing between the base and the collocate or to represent the relation between the base and a support verb. For example, the lexical function *Magn* represents the meaning *intensification*, while the lexical function *Caus*, applied to a base, returns the support verb that represents the causality of the action expressed in the collocation. In a dependency parsing, each word (dependent) is directly associated with its governor in a phrase. In this paper, we show how we combine dependency parsing to extract collocation candidates and a lexical network based on lexical functions to identify the true collocations from the candidates. The candidates are extracted from a French corpus according to 14 dependency relations. The collocations identified are classified according to the semantic group of the lexical functions modeling them. We obtained a general precision (for all dependency types) of 76.3%, with a precision higher than 95% for collocations having certain dependency relations. We also found that about 86% of collocations identified belong to only four semantic categories: *qualification*, *support verb*, *location* and *action/event*.

Keywords: Meaning-Text Theory, lexical function, collocation identification, dependency parsing, lexical network

1 Introduction

A collocation is a type of multiword expression (MWE) in which one of the constituents, the *base*, is chosen freely. The other component, the *collocate*, is chosen by a speaker contingent to the base to express a specific thought. Given the base, there are few possible words for the collocate (usually there is only one) that express the meaning intended by the speaker [1].

For example, in the collocation *strongly condemn*, the base is *condemn*. The collocate *strongly* adds the sense of *intensification* to the base's meaning. The meaning between the base and the collocate can be seen as a *predicate* applied to a *subject* that returns an *object* or *set of objects*. For example: *intensification*(*condemn*) = {*strongly*}.

Beyond the fact that the identification of collocations by a machine is a hard problem, what is the case for any other type of MWE, collocations add the difficulty of having a collocate whose meaning is usually idiomatic or connotative. For example, in the collocation *pay attention*, the collocate *pay* does not have the literal meaning of "*exchanging money for a good or service*".

However, in comparison to other types of MWEs, such as idioms, the different types of predicative meaning between the base and the collocate in a collocation can be categorized. The Meaning-Text Theory's (MTT) lexical functions (LF) [2,1] were created, among other things, to represent each of those meanings. For example, the LF *Magn* represents the meaning *intensification*, which is present in the following collocations: *strongly condemn*, *close shave*, *easy as pie*, *stark naked*, *skinny as a rake*, *rely heavily*, *thunderous applause*, etc. A LF has the form: LF(*base*) = {*collocate*}. For example: Magn (*applause*) = {*thunderous*}.

In some LFs, like Oper_1, subscripts are used to represent the semantic actant of the collocation's base [3]. The subscript used in the LF connecting a base and a collocate identifies if the subject of the verb is the one who is performing the action (subscript = 1), receiving the action (subscript = 2), etc.

Finally, simple LFs, like *Magn*, can be combined to other simple LFs to form complex LFs. Example: AntiMagn (*applause*) = {*scattered*}.

Some methods have been proposed for the identification of collocations. In general, they rely on two main approaches to extract collocation candidates: *n-grams* and *syntactic parsing*. In n-grams based methods, a window containing w words is extracted and all combinations of bigrams or trigrams inside this window are generated. Association measures (e.g. *log-likelihood*) are used to rank those n-grams to create a list of the most probable candidates, as in [4].

In methods based on syntactic parsing, word pairs having some specific syntactic relations are extracted. Usually, those methods rely on three types of word combinations: *verb-noun*, *adjective-noun*, and *noun-noun*. Association measures are used for ranking the best candidates.

In this paper, we propose a method for identifying collocations based on dependency syntactic parsing and we extract candidates belonging to 14 different dependencies (e.g. *subject-verb*, *adjectival modifiers*, *verb-object*, etc.), for French. For filtering the candidates we use a lexical network where words forming collocations are connected by LFs. Finally, we classify the identified collocations according to a semantic perspective for the LFs modeling them.

Our main objective is to precisely identify collocations, instead of having a ranking of probable collocations, as it is the case in methods based on association measures. Moreover, we intend to identify the main semantic groups to which the collocations belong. Finally, we believe that it is important to identify the syntactic dependencies that are more probable to be part in a collocation relation, since the MTT and the LFs are based on dependency syntax.

Another important point is the fact that most work related to distributional semantics deals with paradigmatic relations, such as synonymy and hyperonymy. Syntagmatic relations are mostly ignored [5]. Therefore, we believe that a study showing the semantic distribution of syntagmatic relations, encoded into collocations and LFs, may improve the studies related to distributional semantics.

2 Related work

In general, works on collocation identification are based on a combination of parsing and statistical methods. The majority of those works are based on the ideas proposed by the following precursor works:

- [6] proposed a method for collocation extraction based on the frequency of n-grams (sequence of n words, where $n > 1$);
- [7] tested the use of more sophisticated association measures, based on the theory of mutual information;
- [8] proposed collocation extraction using an annotated corpus with parts of speech (POS) and statistical information, such as the standard deviation and the mean of the distance between words in a sentence.

In the next section, we present some work treating the identification of collocations based on LFs. In Section 2.2, we present a semantic classification for LFs, which we use in the classification of the identified collocations. In Section 2.3, we present the French Lexical Network, which is based on LFs. In Section 2.4, we present the ontology we developed for the representation of LFs and for encoding the French Lexical Network.

2.1 Extraction of collocations based on lexical functions

[9] uses parallel corpora in three languages, English, Spanish and Portuguese, in order to extract collocation candidates. After a preprocessing, a parsing based on Universal Dependencies [10][1] is applied to the corpora, producing files in the CONLL-X [11] format. From those files, candidates pairs having three types of dependency are extracted: *adjective modification* (*amod*) (*adjective-noun*), *nominal modification* (*nmod*) (*noun-noun*) and *verbal object* (*vobj*) (*verb-noun*).

A *t-score* association measure is applied in order to rank the candidates. Only the candidates having a t-score greater than 1 and a frequency higher than 10 are kept. Then, three models are created for each pair of languages (en-es, en-pt and es-pt) using *MultiVec* [12], an implementation of *word2vec* [13] for MWEs, and *BiSkip* [14], a word embeddings model which learns bilingual representations

[1] http://universaldependencies.org/introduction.html

using aligned corpora. Those models are applied in the identification of equivalent collocations between a source and a target language.

In [15], Kolesnikova evaluates 68 supervised classification algorithms (e.g. *Bayes Net*, *Bagging*, *AdaBoostM1*, etc.), for the classification, into different lexical functions, of several Spanish collocations having the pattern *verb-noun*.

For the construction of the training sets, *verb-noun* pairs are extracted automatically from a corpus. The most frequent pairs are selected and those representing collocations are manually annotated with their respective LFs. The pairs that are not collocations are annotated as FWC - (Free word combination).

For each pair *verb-noun*, hyperonyms are extracted from the Spanish WordNet. If no hyperonym is found for a verb or for a noun in a pair, it is excluded from the list. A training set is created for each LF. As a result, for example, the *Bayesian Logistic Regression* algorithm is the most efficient in the identification of collocations modeled by the FL $Oper_1$.

[16] uses *FrameNet* [17] to extract collocations having a support verb. Each frame is associated with lexical units that evoke it. For example, the *Judgment* frame is evoked by lexical units (LU) such as *accusation*, *critique*, etc. A LU that evokes a frame is a *target.* Text corpora associated with FrameNet are annotated with frame element and target names, such as *support verb*, *controller*, etc.

The method of [16] consists in automatically browsing an annotated corpus with frames and target words to locate the annotated verbs such as *Supp* (support verbs) and the target words associated with those support verbs. Each pair (*Supp*, *target*) is a collocation having a support verb and is associated with the LF *Oper*. A heuristic is employed in order to determine the *Oper* function's index: for example, if the subject of the support verb is an agent, a person, etc., the index of *Oper* is *1*, so the function is $Oper_1$.

[18] presents a method where classification algorithms are used to assign collocations to the LF that model them. Their method is based on the *K-nearest neighbor algorithm*, which can be used when prototypical collocations are defined for the LF that model them.

From the collocations given as examples, the algorithm can automatically classify other collocations. [18] present two other classification methods, based on *naive Bayes networks* to automatically assign LFs to collocations.

[4] presents *Colex*, a tool based on LFs that combines symbolic and statistical approaches for extracting terminological collocations for English. As in [15], the extracted collocations follow the pattern *verb-noun* and three types of grammatical relations: subj. + verb (*The program executes*), verb + direct object: (*[to] close a file*) and verb + indirect object (*[to] load into memory*).

The method for the extraction of candidates is the matching between rules that follow these relation types and the syntactic tree obtained for a parser applied to the sentences of an English corpus. Two association measures, *mutual information* and *log-likelihood* are employed in order to increase the precision.

2.2 Semantic perspective for lexical functions

In [19], different classification of lexical functions are presented, which are called "perspectives". In the semantic perspective (SP), LFs are classified into ten main classes. We added two more in our ontology, *supportVerb* and *semanticallyEmptyVerb*. These two added classes are not semantic in a strict sense because they represent "semantically empty verbs". However, since a great number of collocations are formed by these types of verbs, we decided to add these two classes.

The 12 main classes are presented in Table 1.

Table 1: Semantic perspective classes for lexical functions

actionEvent	causativity	elementSet	equivalence
location	opposition	participants	phaseAspect
qualification	semanticallyEmptyVerb	supportVerb	utilizationForm

Some of the SPs are subdivided into subclasses. For example, phase/aspect is divided in *preparation*, *start*, *continuation*, *duration*, *reiteration*, etc. Examples of collocations and their respective LFs classified as phase/aspect:

- *déployer ses ailes* (*stretch the wings*): PreparReal$_1$(*ailes*) = {*déployer*} - (*phase/preparation*);
- *adopter une attitude* (*adopt an attitude*): IncepOper$_1$(*attitude*) = {*adopter*} - (*phase/start*);

Examples of LFs and collocations for the other SPs:

- action/event - *jouer du piano* (*play the piano*): Real$_1$(*piano*) = {*jouer*};
- causativity - *par politesse* (*out of politeness*): Propt(*politesse*) = {*par*};
- location - *sur le lit* (*on the bed*): Loc$_{in}$(*lit*) = {*sur*};
- opposition - *bref délai* (*short delay*): AntiMagn(*délai*) = {*bref*}. In this example, the opposition is not between the base (*délai*) and the value (*bref*). Instead, it is an opposition of the intensification relation: Magn(*délai*) = {*long*} → AntiMagn(*délai*) = {*bref*};
- participants - *siège vacant* (*vacant seat*): A$_2$NonReal$_1$(*siège*) = {*vacant*};
- qualification - *politique efficace* (*effective policy*): Ver(*politique*) = {*efficace*};
- semant. empty verb - *être victime* (*to be a victim*): Pred(*victime*) = {*être*};
- support verb - *courir le risque* (*take the risk*): Oper$_1$(*risque*) = {*courir*};
- utilization/form - *par avion* (*by plane*): Instr(*avion*) = {*par*}.

The SPs *element/set* and *equivalence* represent paradigmatic relations. Therefore, there are no collocations classified in those classes. Examples of LFs and paradigmatic relations for those classes:

- element/set (hyperonymy, meronymy, etc.) - *chat/félin* (*cat/feline*): Hyper(*chat*) = {*félin*};
- equivalence (syntactic conversion) - *acclamation/acclamer* (*acclamation/to acclaim*): V$_0$(*acclamation*) = {*acclamer*};

2.3 The French Lexical Network

The French Lexical Network (FLN) [20] is, to our knowledge, the only lexical network based on LFs. It has been built manually by a lexicographic team of around 15 people, as part of the project *RELIEF*[2]. Lexicographic strategies used to extract linguistic information from corpora and build the network are based on the Explanatory Combinatorial Lexicology [21]. It makes extensive use of the *Digital Thesaurus of the French Language*[3] (in French, *Trésor de la Langue Française informatisé*) [22] as a lexical database for lexicographic information.

In the FLN, each word is represented together with its possible meanings, each meaning being a lexeme of a word. Each lexeme is represented by the word form and a combination of Romans and Arabic numbers. For example, the word "*vêtement*" (*clothing*) has five meanings inside the FLN: $vêtement_{I.1}$, $vêtement_{I.2}$, $vêtement_{II}$, $vêtement_{III.1}$ and $vêtement_{III.2}$

In the FLN, the paradigmatic and syntagmatic relations between lexemes are represented by LFs. For example, between "*petit*" (*small*) and "*grand*" (*big*) there is the LF *Anti* (antonymy) connecting them. Or between "*très*" (*very*) and "*grave*" (*serious*) there is the LF *Magn* connecting them.

2.4 The FLN in an ontology format

We have developed a lexical ontology, called *lexical function ontology model* (*lexfom*) [23] to represent LFs and lexical relations based on LFs and we have applied this ontology in the transformation of the FLN to an ontology format.

Having the FLN in an ontology format facilitates the access of its content, allow its combination with other lexical resources on the semantic web and sets a standard that can be followed by other lexical networks based on LF for other languages. Lexfom has four modules:

- *Lexical functions representation* (*lfrep*): represents the individual properties of a LF, such as syntactic actants, if it is a simple or complex LF, etc.;
- *Lexical functions relation* (*lfrel*): represents a syntagmatic or paradigmatic relation, indicating the base and the value of the relation;
- *Lexical functions family* (*lffam*): represents a syntactic classification of LFs;
- *Lexical functions semantic perspective* (*lfsem*): represents the semantic perspectives for LFs, presented in the Section 2.2.

The FLN in ontology format contains about $8,000$ different syntagmatic and about $46,000$ different paradigmatic relations between lexemes. Moreover, our ontology represents about 600 different LFs, where 500 are complex LFs and 100 are simple LFs. In this ontology, LFs like, for example, $Oper_1$, $Oper_2$ and $Oper_3$ are considered as different LFs.

He have then created a Java API to access the FLN in an ontology format. For the present paper, the relevant API's functions are:

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

[3] http://www.atilf.fr/spip.php?rubrique77

- searchSyntagRelation(gov, dep) → LF(*base*, *collocate*): the pair (*governor*, *dependent*) (or any pair of words) are searched in the part of the ontology representing the syntagmatic relations (SR). If there is at least one SR between them, the result of the query returns the LF connecting them and which one is the base and the collocate in this relation.
  Example: searchSyntagRelation(*pose*, *questions*) → Oper_1(*questions*, *pose*);
- searchSemPerspective (LF) → semanticPerspective : returns the semantic perspective of a LF. Example: searchSemPerspective(Oper_1) → *supportVerb*.

3 Methodology

We present our method for extracting collocation candidates from a corpus and identifying them as true collocations. Figure 1 presents our methodology. The sequence of steps is as follows:

- Pre-processing of the corpus: tokenization, POS-tagging and lemmatization;
- A dependency parsing is applied and a file in the CONLL-X [11] format is generated (next two steps);
- The pairs (*gov*, *dep*) are extracted from the CONLL-X file according to some specific dependency types;
- The pairs extracted are matched against our ontology representing the FLN and a list of collocation candidates is generated;
- All the candidates are manually analyzed and a collocations list is generated;
- The collocations are classified accordingly to the SP of their LFs.

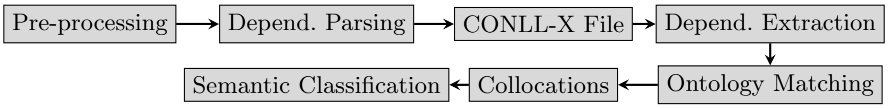

Fig. 1: The Pipeline for the extraction of candidates and the identification and classification of collocations.

In Section 3.1, we present the steps from the dependency parsing to the CONLL-X file generation. In Section 3.2, we show how the ontology matching is used to generate a list of collocations. In Section 3.3, we show how the identified collocations are semantically classified.

3.1 Use of dependency syntax to extract candidates

After the preprocessing, a dependency parser (MaltParser[4]) is applied. According to [24], the MaltParser's accuracy for French is around 89%. They showed

[4] http://www.maltparser.org/

that MaltParser is about 1 to 2 p.p. less precise than Berkeley Parser[5] and MSTParser[6], but it is about 12 to 14 times faster. The parsing generates a file in the CONLL-X format, where each token in a phrase is represented by a row containing 10 columns. The most important for this work are:

- an ID (position) of the token in the phrase;
- surface form; – lemma; – POS; – dependency type;
- dependency head ID in the phrase (zero if the token is the root in the phrase).

We extract the collocation candidates from the CONLL-X file. We extract only word pairs having between them the following dependency relations [25]:

- a_obj: argument introduced by "à" - *à fond* (*thoroughly*);
- arg: argument (expressions connected by a preposition) - *comme tout* (*like any other*);
- ats: predicative adjective or nominal over the subject (*attribut du sujet*) - *être victime* (*to be a victim*);
- aux_caus: causative auxiliary verb - *faire dégager* (*to make clear*);
- aux_tps: tense auxiliary verb (*auxiliare de temps*) - *avoir vu* (*have seen*);
- coord: links a coordinator to the immediately preceding conjunct - in the phrase "*le garçon et la fille*" (*the boy and the girl*), there is a *coord* relation between the coordinator *et* and the preceding conjunct *garçon*;
- de_obj: argument introduced by "de" - *souvient de* (*remembers*);
- dep: unspecified dependency - *très grave* (*very serious*);
- dep_coord: links a conjunct to the previous coordinator - in the phrase above, given as example for the dependency *coord*, there is a *dep_coord* relation between the conjunct *fille* and the previous coordinator *et*;
- mod: modifiers (adjectival, nominal and adverbial) other than relative phrases - *politique véritable* (*true policy*);
- mod_rel: links a relative pronoun's antecedent to the verb governing the relative phrase - *série qui plaît* (*series that pleases*), mod_rel(*série*, *plaît*);
- obj: object of a verb - *traiter les maladies* (*to treat diseases*);
- p_obj: arg. introduced by another preposition - *sur la table* (*on the table*);
- suj: subject of a verb - *le bateau naviguait* (*the boat was sailing*).

Most of the recent work based on parsing use the tag sets provided by the Universal Dependency (UD) project[10]. The aim of this project is to have a tag set of dependencies that are the same for all languages and facilitate the sharing and comparison of information.

The dependencies used in this paper are not the same as the dependencies used in UD. Our method is based on a lexical network based on LFs, which, for the moment, only exists for the French language. For MaltParser, the only French language model available[7] was developed before the UD creation.

[5] http://www.eecs.berkeley.edu/~petrov/berkeleyParser

[6] http://mstparser.sourceforge.net

[7] http://maltparser.org/mco/french_parser/fremalt.html

The use of a dependency parser can improve precision in the identification of collocations for two reasons. The first one, the relation between a base and a collocate in a collocation is a dependency relation. The second one, a parser can identify with good accuracy dependency relations between words that are some words apart, what a method based on a window of n words may fail to identify.

3.2 Use of the FLN to filter candidates

From the CONLL-X files, each pair (*gov*, *dep*) having one of the 14 dependencies types where matched against our ontology representing the syntagmatic relations extracted from the FLN. We used the lemmas of each *gov* and *dep* in the search.

A match means that the pair having a dependency relation in a phrase also has a syntagmatic relation in the French language and is possibly a collocation. Our search in the ontology returns also the LF connecting the pair. This allows us not only to identify collocations but also identify the LF modeling the relation.

For the positive matches against our ontology, we keep the following information: the surface forms of the *gov* and *dep* words, their lemmas, the LF connecting them in the ontology, the information about which word is the base and which one is the collocate and the syntactic dependency between them in the text.

Due to the lack of resources, we could not make the analysis on the negative matches, that is, the true collocations in the corpus that are not present in the FLN. For this type of analysis, we would need a French corpus annotated with collocations and LFs, what is not yet available. For the same reason we measure our results by precision only, without calculating the recall.

For some pairs, we can have more than one LF connecting them. As an example, the pair (*adopter*, *politique*):

- IncepOper_1 (*politique*) = {*adopter*};
- Real_{1-I} (*politique*) = {*adopter*};

3.3 Semantic classification of collocations

For each match, we search for the SP of each LF modeling the collocation. Since LFs can be complex, i.e., formed by the combination of two or more simple LFs, a LF can have more than one SP and as a consequence, the collocation modeled by this LF will be classified in more than one semantic group.

For example, the complex LF $\textit{FinReal}_1$ is composed by the LF *Fin*, whose SP is "*actionEvent*", subclass "*disparation/existencial cease*", and by the LF $\textit{Real}_1$, whose SP is "*actionEvent*", subclass "*utilization/typical operation*". Therefore, the French collocation "*abandoner la politique*" (*to quit politics*), which is modeled by the LF $\textit{FinReal}_1$ ($\textit{FinReal}_1$ (*politique*) = {*abandoner*}), will be classified in two semantic subclasses of *actionEvent.* This feature can be useful, for example, in a multi-labeling classification system that uses collocations to identify multi-classes of a phrase or document.

In the case of pairs having more than one LF, we choose to classify them according to the first LF returned.

4 Experiments and analysis of results

We use and exploit the EuroSense corpus [26] in our experiments. EuroSense is a multilingual parallel corpus, containing sentences in 21 languages. We extract all French phrases (about 1.8 million) contained in EuroSense.

The next step is the preprocessing: all phrases are segmented and POS tagged using the Apache OpenNLP[8] (OpenNLPSegmenter and OpenNlpPosTagger). Then the lemmatization is performed with DKPRO LanguageToolLemmatizer[9]. MaltParser[10] is used as a dependency parser. Finally, we use DKPRO to generate a file containing all phrases in the CONLL-X format.

We perform two types of analysis. In the first one, we measure the collocation identification precision by dependency type. The objective is to evaluate for which dependencies we can obtain more collocations with higher precision. In the second one, we count how many collocations are identified by SP. The objective is to evaluate the most common semantic relations in French collocations.

We could not calculate the recall since this is a first study of this type for French and there is no corpus annotated with French collocations and LFs.

4.1 Collocations classified by syntactic dependencies

Table 2 shows the candidates extracted and collocations correctly identified by syntactic dependency. For each type of dependency, we have the total number of candidates extracted, the total number of true collocations and the precision (number of true collocations / number of candidates). We extracted $43,629$ collocation candidates and $33,273$ were identified as true collocations.

Since there are no available corpora annotated with collocations and LFs for the French, we calculated the precision manually, over all the collocation candidates extracted: each pair is observed and a human annotator decides if a pair is or not a collocation. Although we have many candidates, the number of different individual collocations is low since many are repeated many times, what facilitates the manual annotation.

Except for the *arg* dependency, that produced only seven collocations, we had the highest precision with the pairs having the *obj* (96.9%), *a_obj* (98.3%) and *p_obj* (95.6%) dependencies. In other words, the relations where the *dependent* is a *governor*'s object had the best precision overall.

The most similar work is the one from [9], presented in Section 2.1. Besides the fact that they deal with the extraction of bilingual collocations, what is a harder task, the difference is that they only use three dependencies: *amod*, *nmod* and *vobj*, which are less likely to produce errors because 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*.

[8] https://opennlp.apache.org/

[9] https://dkpro.github.io/

[10] http://www.maltparser.org/

Table 2: Precision for the extraction of collocations by syntactic dependency.

dependency	**nr. candidates**	**nr. true coll.**	**precision**
mod	20625	14240	0.690
dep	14015	11532	0.823
obj	4869	4720	0.969
suj	1249	688	0.551
mod_rel	1179	888	0.753
coord	605	442	0.731
ats	400	346	0.865
a_obj	300	295	0.983
dep_coord	246	13	0.053
p_obj	90	86	0.956
aux_tps	37	15	0.405
arg	7	7	1.000
aux_caus	7	1	0.143
de_obj	0	0	0
Total	**43629**	**33273**	**0.763**

To our knowledge, there is no other work, besides ours, on the extraction of French collocations using syntactic dependency or the FLN.

In general, we expected to have a 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:

- errors connected to the verb "*être*" (*to be*). For example, in the phrase "*on est pêcheur de père en fils*" (lit. *we are fishermen from father to son*) the verb "*est*" is not syntactically dependent on "*père*". However, the parser found a dependency between them. And since "*être père*" (lit. *to be a father*) is a collocation in French ($Oper_{12}$ (*père*) = {*être*}), we had a false positive;
- errors connected to the verb "*avoir*" (*to have*). For example, in the sentence "*...transport des animaux vivants ait été décidée, même si nous aurions préféré...*" (lit. *transport of live animals has been decided, even if we would have preferred...*), the parser found a dependency (*mod*) between "animaux" and "aurions", what is false. It was considered a collocation since, in the FLN, there is the following relation: $Real_1$ ($animal_{I.2}$) = {$avoir_{I.1}$};
- errors with the verb "*pouvoir*" (*to be able to*). In many phrases, this verb was considered a noun and we had some false positives because, for example, "*tenir le pouvoir*"(*to keep the power*) is a collocation;
- the word "*car*" (*because*), which is also a colloquial word for "*autocar*" (*coach* or *bus*), was often tagged as a noun, appearing incorrectly as a collocation in expressions like "*dans le car*" (*inside the car*). For the pairs having the dependency *dep_coord*, "*dans le car*" was the most common candidate and this explains why candidates in this group had a low precision.

The first three types of errors respond for 95.7% of the errors, in the following proportion: "*pouvoir*" errors = 35.1%; "*avoir*" errors = 31.1%; "*être*" errors = 29.5%. The "*car*" errors correspond to 3.9% of the errors, while only 0.4% (43/10, 356) are due to other types of errors.

This error distribution means that it would be easy to decrease the error rate by concentrating an extra effort in dealing with dependencies having few types of words or having a more detailed analysis on the RLF's syntagmatic relations where those specific words appear.

4.2 Collocations classified by semantic perspectives

The 33, 273 identified collocations were classified by the semantic perspective of their lexical functions. Since some collocations are modeled by complex LFs, they can be classified into more than one SP, which gives 35, 243 different instances of SP classifications. Table 3 shows the number of collocations identified, classified by each SP. For the SPs *actionEvent*, *phaseAspect*, *qualification* and *equivalence* we have also identified collocations for their subclasses.

Table 3: Number of collocations identified by semantic perspective.

semantic perspective	**n. coll.**	**example**	**lexical function**
qualification	11983		
intensity	5988	*très grave*	Magn(*grave*) = {*très*}
judgment	5940	*aller bien*	Bon(*aller*) = {*bien*}
inten+judg	55	*trop rapide*	AntiBon+Magn(*rapide*) = {*trop*}
supportVerb	8620	*donner un coup*	Oper_1(*coup*) = {*donner*}
location	6332	*dans le pays*	Loc_{in}(*pays*) = {*dans*}
actionEvent	3428		
utilizationTypOper	2846	*mettre l'accent*	Real_1(*mettre*) = {*accent*}
creation	580	*faire voler*	Caus(*voler*) = {*faire*}
disparExistCease	121	*l'avion atterrit*	FinFact_0(*avion*) = {*atterrir*}
imminence	5	*la tempête vient*	ProxFunc_0(*tempête*) = {*venir*}
semanticallyEmptyVerb	1693	*être similaire*	Pred(*similaire*) = {*être*}
utilizationForm	1410	*à vélo*	Instr(*vélo*) = {*à*}
phaseAspect	729		
start	632	*devenir mère*	IncepPred(*mère*) = {*devenir*}
continuation	77	*rester debout*	ContPred(*debout*) = {*rester*}
preparation	20	*déployer les ailes*	PreparReal_1(*ailes*) = {*déployer*}
semanticOpposite	647	*mauvais sens*	AntiBon(*sens*) = {*mauvais*}
causativity	362	*par politesse*	Propt(*politesse*) = {*par*}
participants (actants)	39	*cigarette allumé*	A_1Fact_0(*cigarette*) = {*allumer*}
Total	35243		

The most common SP for collocations were *qualification* (33.9%), *supportVerb* (24.4%), *location* (17.9%) and *actionEvent* (9.7%). These four SPs represent about 86% of all collocations identified.

The collocations classified into the group *qualification* were divided almost equally between the sub-groups *intensity* and *judgment*. As examples of collocations belonging to this group we have:

- "*crime odieux*" (*heinous crime*): Magn(*crime*) = {*odieux*} - (intensity)
- "*chiffre exact*" (*exact number*): Ver(*chiffre*) = {*exact*} - (judgment)

Among the collocations classified as *supportVerb*, the most common are the ones having LFs belonging to the $Oper_1$ "family". For example:

- "*poser une question*" (*to ask a question*): Oper_1(*question*) = {*poser*}

Almost all the collocations classified into the group *location* are modeled by the LF Loc_{in}. Examples of collocations belonging to this group:

- "*dans le pays*" (*in the country*): Loc_{in} (*pays*) = {*dans*}
- "*en semaine*" (*during the week*): ${Loc_{in}}^{time}$ (*semaine*) = {*en*}

For the collocations classified as *actionEvent*, the most common sub-groups were *utilizationTypicalOperation* (u/t.o) and *creation*. For example:

- "*donner la forme*" (*give shape*): $\mathrm{CausFunc}_1$ (*forme*) = {*donner*} - (creation)
- "*mettre l'accent*" (*to accentuate*): Real_1 (*accent*) = {*mettre*} - (u/t.o)

This type of analysis and classification by semantic group could be useful in, for example, those applications related to distributional semantics:

- sentiment analyses: the identification of collocations in the semantic group *qualification* in a phrase may be useful in the identification of the sentiment expressed. Either positive, if the collocation is modeled by the LFs *Bon* or *Ver*, or negative, if modeled by the LFs *AntiBon* or *AntiVer*;
- text classification: the presence of a specific collocation in a specific semantic group may help in the identification of the topic of a sentence.

5 Conclusion and future work

In this paper, we presented a method for identifying collocations from a corpus. To decide which candidate is a true collocation, we performed a search in a lexical network based on lexical functions, the French Lexical Network (FLN), using a lexical ontology that we developed to access the information in the FLN.

More than searching for collocations, we also obtained the lexical functions connecting the base and the collocate and each lexical function's semantic(s) perspective(s). We calculated the precision of the identified collocations for each type of dependency and we compared the total number of collocations by semantic perspective. For some dependency relations, we had a precision higher than 95%. And we analyzed that, if we deal properly with punctual issues, like some parsing errors, we can achieve a precision close to 100%.

As future work, we intend to combine our method with machine learning algorithms based on word embeddings. Machine learning works well for collocations having a frequency higher than a certain threshold. We believe that the use of a lexical network where the relations of collocations (syntagmatic relations) were manually annotated by lexicographers will help in the identification of less frequent collocations, improving the performance.

Also, we expect to extend this method to a lexical network based on lexical functions for English (under construction), and apply it to the translation of collocations for the pair English-French. Finally, we intend to create a French corpus annotated with collocations and lexical functions, to allow future automatic evaluation of precision and recall in the identification of collocations.

References

1. Igor Mel'čuk. Collocations and Lexical Functions. *Phraseology. Theory, Analysis and Applications*, pages 23–53, 1998.
2. Igor Mel'čuk. *Vers une linguistique sens-texte*. Collège de France Paris: Leçon inaugurale. Collège de France, 1997.
3. Igor Mel'čuk. *Actants in semantics and syntax II: actants in syntax*, volume 42, pages 247–291. de Gruyter.
4. Brigitte Orliac. Colex. Un outil d'extraction de collocations spécialisées basé sur les fonctions lexicales. *Terminology*, 12:261–280, 2006.
5. Magnus Sahlgren. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-dimensional Vector Spaces*. SICS Dissertation Series. Department of Linguistics, Stockholm University, 2006.
6. Yaacov Choueka. Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In Christian Fluhr and Donald E. Walker, editors, *RIAO*, pages 609–624. CID, 1988.
7. Kenneth Ward Church and Patrick Hanks. Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990.
8. Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Comput. Linguist.*, 22(1):1–38, March 1996.
9. Marcos Garcia, Marcos García-Salido, and Margarita Alonso-Ramos. 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 2017. Association for Computational Linguistics.
10. Ryan T. McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal Dependency Annotation for Multilingual Parsing. In *ACL (2)*, pages 92–97. The Association for Computer Linguistics, 2013.
11. Sabine Buchholz and Erwin Marsi. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

12. Alexandre Berard, Christophe Servan, Olivier Pietquin, and Laurent Besacier. Multivec: a Multilingual and Multilevel Representation Learning Toolkit for NLP. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).
13. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.
14. Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.
15. Olga Kolesnikova. *Automatic Extraction of Lexical Functions.* PhD thesis, 2011. PhD Thesis directed by Alexander Gelbukh, Instituto Politecnico Nacional – Centro de Investigacion en Computacion, Mexico, DF.
16. Margarita Alonso Ramos, Owen Rambow, and Leo Wanner. Using Semantically Annotated Corpora to Build Collocation Resources. In *Proceedings of LREC*, pages 1154–1154, 2008.
17. Charles J. Fillmore. *Scenes-and-Frames Semantics.* Number 59 in Fundamental Studies in Computer Science. North Holland Publishing, 1977.
18. Leo Wanner, Bernd Bohnet, and Mark Giereth. What is Beyond Collocations? Insights from Machine Learning Experiments. In Cristina Onesti Elisa Corino, Carla Marello, editor, *Proceedings of the 12th EURALEX International Congress*, pages 1071–1087, Torino, Italy, sep 2006. Edizioni dell'Orso.
19. Anne-Laure Jousse. *Modèle de structuration des relations lexicales fondé sur le formalisme des fonctions lexicales.* PhD thesis, 2010. Directed by Sylvain Kahane et Alain Polguere, Université de Montréal et Université Paris Diderot (Paris 7).
20. Veronika Lux-Pogodalla and Alain Polguère. Construction of a French Lexical Network: Methodological Issues. In *First InternationalWorkshop on Lexical Resources, WoLeR 2011*, pages 54–61, Ljubljana, Slovenia, August 2011.
21. Igor Mel'čuk, André Clas, and Alain Polguère. *Introduction à la lexicologie explicative et combinatoire.* Duculot, Louvain-la-Neuve (Belgique), 1995.
22. Jacques Dendien and Jean-Marie Pierrel. Le trésor de la langue française informatisé: un exemple d'informatisation d'un dictionnaire de langue de référence. *Traitement Automatique des Langues (TAL)*, 44:11–37, 2003.
23. Alexsandro Fonseca, Fatiha Sadat, and François Lareau. Lexfom: a lexical functions ontology model. In *Proceedings of the Fifth Workshop on Cognitive Aspects of the Lexicon (CogALex), COLING*, pages 145–155, Osaka, 2016.
24. Marie Candito, Joakim Nivre, Pascal Denis, and Enrique Henestroza Anguiano. Benchmarking of Statistical Dependency Parsers for French. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 108–116, Stroudsburg, PA, USA, 2010. ACL.
25. Marie Candito, Enrique Henestroza Anguiano, and Djamé Seddah. A Word Clustering Approach to Domain Adaptation: Effective Parsing of Biomedical Texts. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 37–42, Vancouver, Canada, 2011.
26. Alessandro Raganato Claudio Delli Bovi, José Camacho Collados and Roberto Navigli. 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, 2017.