Parsing and MWE Detection: Fips at the PARSEME Shared Task

Vasiliki Foufi and **Luka Nerima** and **Éric Wehrli**
LATL-CUI, University of Geneva
7 route de Drize
CH-1227 Carouge, Switzerland
{vasiliki.foufi, luka.nerima, eric.wehrli}@unige.ch

Abstract

Identifying multiword expressions (MWEs) in a sentence in order to ensure their proper processing in subsequent applications, like machine translation, and performing the syntactic analysis of the sentence are interrelated processes. In our approach, priority is given to parsing alternatives involving collocations, and hence collocational information helps the parser through the maze of alternatives, with the aim to lead to substantial improvements in the performance of both tasks (collocation identification and parsing), and in that of a subsequent task (machine translation). In this paper, we are going to present our system and the procedure that we have followed in order to participate to the open track of the PARSEME shared task on automatic identification of verbal multiword expressions (VMWEs) in running texts.

1 Introduction

Multiword expressions (MWEs) are lexical units consisting of more than one word (in the intuitive sense of 'word'). There are several types of MWEs, including idioms (*a frog in the throat*, *break a leg*), fixed phrases (*per se*, *by and large*, *rock'n roll*), noun compounds (*traffic light*, *cable car*), phrasal verbs (*look up*, *take off*), etc. While easily mastered by native speakers, their detection and/or their interpretation pose a major challenge for computational systems, due in part to their flexible and heterogeneous nature.

In our research, MWEs are categorized in five subclasses: compounds, discontinuous words, named entities, collocations and idioms. While the first three are expressions of lexical category (N, V, Adj, etc.) and can therefore be listed along with simple words, collocations and idioms are expressions of phrasal category (NPs, VPs, etc.). The identification of compounds and named entities can be achieved during the lexical analysis, but the identification of discontinuous words (e.g. particle verbs or phrasal verbs), collocations and idioms requires grammatical data and should be viewed as part of the parsing process.

In this paper, we will primarily focus on collocations, roughly defined as arbitrary and conventional associations of two words (not counting grammatical words) in a particular grammatical configuration (adjective-noun, noun-noun, verb-object, etc.) and especially on the categories of verbal collocations defined in the framework of the PARSEME shared task.

Section 2 will give a brief review of MWEs and previous work. Section 3 will describe how our system handles MWEs, the way they are represented in its lexical database and will also be concerned with the treatment of collocation types which present a fair amount of syntactic flexibility (e.g. verb-object). For instance, verbal collocations may undergo syntactic processes such as passivization, relativization, interrogation and even pronominalization, which can leave the collocation constituents far away from each other and/or reverse their canonical order. Section 4 will present the modifications made in order to adapt our system to the requirements of the shared task and the section 5 the evaluation and results.

2 Multiword expressions: a brief review of related work

The standard approach in dealing with MWEs in parsing is to apply a 'words-with-spaces' preprocessing step, which marks the MWEs in the input sentence as units which will later be integrated as

Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), pages 54–59,
Valencia, Spain, April 4. ©2017 Association for Computational Linguistics

single blocks in the parse tree built during analysis (Brun, 1998; Zhang and Kordoni, 2006). This method is not really adequate for processing collocations. Unlike other expressions that are fixed or semi-fixed, several collocation types do not allow a 'words-with-spaces' treatment because they have a high morphosyntactic flexibility. On the other hand, Alegria et al. (2004) and Villavicencio et al. (2007) adopted a compositional approach to the encoding of MWEs, able to capture more morphosyntactically flexible MWEs. Alegria et al. (2004) showed that by using a MWE processor in the preprocessing stage, a significant improvement in the POS tagging precision is obtained. However, as argued by many researchers, e.g. (Heid, 1994; Seretan, 2011; Wehrli and Nerima, 2013), collocation identification is best performed on the basis of parsed material. This is due to the fact that collocations are co-occurrences of lexical items in a specific syntactic configuration. Additionally, Nasr et al. (2015) have developed a joint parsing and MWE identification model for the detection and representation of ambiguous complex function words. Constant and Nivre (2016) developed a transition-based parser which combines two factorized substructures: a standard tree representing the syntactic dependencies between the lexical elements of a sentence and a forest of lexical trees including MWE identified in the sentence.

3 The Fips parser

Our system is a multilingual parser, available for several languages, i.e. French, English, German, Italian, Spanish, Modern Greek, Romanian and Portuguese (Wehrli, 2007; Wehrli and Nerima, 2015). It relies on generative grammar concepts and is basically made up of a generic parsing module which can be refined in order to suit the specific needs of a particular language. It is a constituent parser that functions as follows: it scans an input string from left to right, without any backtracking. The parsing algorithm, iteratively, performs the following three steps:

- get the next lexical item and project the relevant phrasal category
  X → XP, where X ∈ {V, N, Adj, ... }

- merge XP with the structure in its left context (the structure already built);

- (syntactically) interpret XP, triggering procedures
  - to build predicate-argument structures
  - to create chains linking preposed elements to their trace
  - to find the antecedent of (3rd person) personal pronouns

The parsing procedure is a one pass (no preprocessing, no post-processing) scan of the input text, using rules to build up constituent structures and (syntactic) interpretation procedures to determine the dependency relations between constituents (grammatical functions, etc.), including cases of long-distance dependencies. One of the key components of the parser is its lexicon which contains detailed morphosyntactic and semantic information, selectional properties, valency information, and syntactico-semantic features that are likely to influence the syntactic analysis.

3.1 The lexicon

The lexicon is built manually and contains fine grained information required by the parser. It is organized as a relational database with four main tables:

- **words**, representing all morphological forms (spellings) of the words of a language, grouped into inflectional paradigms;

- **lexemes**, describing more abstract lexical forms which correspond to the syntactic and semantic readings of a word (a lexeme corresponds roughly to a standard dictionary entry);

- **collocations**, which describe multi-word expressions combining two lexical items, not counting function words;

- **variants**, which list all the alternative written forms for a word, e.g. the written forms of British English vs American English, the spellings introduced by a spelling reform, presence of both literary and modern forms in Greek, etc.

3.2 Representation of MWEs in the lexicon

In the introduction, we mentioned that in our research the MWEs are categorized in five subclasses, i.e. compounds, discontinuous words, named entities, collocations and idioms. Let's see how they are represented in the lexical database.

Compounds and named entities are represented by the same structure as simple words. An entry describes the syntactic and (some) semantic properties of the word: lexical category (POS), type (e.g. common noun, auxiliary verb), subtype, selectional features, argument structure, semantic features, thematic roles, etc. Each entry is associated with the inflectional paradigm of the word, that is all the inflected forms of the word along with the morphological features (number, gender, person, case, etc.). The possible spaces or hyphens of the compounds are processed at the lexical analyzer level in order to distinguish those that are separators from those belonging to the compound.

Discontinuous words, such as particle verbs or phrasal verbs, are represented in the same way as simple words as well, except that the orthographic string contains the bare verb only, the particle being represented separately in a specific field. The benefit of such an approach is that the phrasal verb inherits the inflectional paradigm of the basic verb. For agglutinative languages, a lexical analyzer will detect and separate the particle from the basic verb.

Collocations are defined as associations of two lexical units (not counting function words) in a specific syntactic relation (for instance adjective - noun, verb - noun (object), etc.). A lexical unit can be a word or a collocation. The definition is therefore recursive and enables to encode collocations that have more than two words. For instance, the French collocation *tomber en panne d'essence* ('to run out of gas') is composed of the word *tomber* and the collocation *panne d'essence*. Similarly, the English collocation *guaranteed minimum wage* is composed of the word *guaranteed* and collocation *minimum wage*.

In addition to the two lexical units, a collocation entry encodes the following information: the citation form, the collocation type (i.e. the syntactic relation between its two components), the preposition (if any) and a set of syntactic frozenness constraints.

For the time being, we represent idioms like collocations, with more restriction features (cannot passivize, no modifiers, etc.) and are, therefore, stored in the same database table. Reducing idioms to collocations with specific features, though convenient and appropriate for large classes of idioms, is nevertheless not general enough. In particular, it does not allow for the representation of idioms with fixed phrases, such as *to get a foot in the door*.

3.3 Parsing and collocations

3.3.1 Collocation identification mechanism

The collocation identification mechanism is integrated in the parser. In the present version of the parser, collocations, if present in the lexicon, are identified in the input sentence during the analysis of that sentence, rather than at the end. In this way, priority can be given to parsing alternatives involving collocations. Thus collocational information helps the parser through the maze of alternatives as shown in Wehrli (2014). To fulfil the goal of interconnecting the parsing procedure and the identification of collocations, we have incorporated the collocation identification mechanism within the constituent attachment procedure (see next section). Our parser, like many grammar-based parsers, uses left attachment and right attachment rules to build respectively left subconstituents and right subconstituents. The grammar used for the computational modelling comprises rules and procedures. Attachment rules describe the conditions under which constituents can combine, while procedures compute properties such as long-distance dependencies, agreement, control properties, argument-structure building, and so on.

3.3.2 Treatment of collocations

The identification of a collocation occurs when the second lexical unit of the collocation is attached, either by means of a left attachment rule (e.g. adjective-noun, noun-noun) or by means of a right-attachment rule (e.g. noun-adjective, noun-prep-noun, verb-object). In the example *Paul took up a new challenge*, when the parser reads the noun *challenge* and attaches it (along with the prenominal adjective) as complement of the incomplete direct object of the verb *take up*, the identification procedure considers iteratively all the governing nodes of the attached noun and checks whether the association of the lexical head of the governing node and the attached element constitutes an entry in the collocation database. The process stops at the first governing node of a major category (noun, verb or adjective). In our example, going up from *challenge*, the process stops at the verb *take up*. Since *take up - challenge* is an entry in the collocation database and its type (verb-object) corresponds to the syntactic configuration, the identification process succeeds.

As already pointed out, in several cases the two constituents of a collocation can be very far apart, or do not appear in the expected order. For instance, verb-object collocations may undergo syntactic processes such as passivization, relativization, interrogation and even pronominalization, which can leave the collocation constituents far away from each other and/or reverse their canonical order.

In passive constructions, the direct object is promoted to the subject position leaving a trace, i.e. an empty constituent in the direct object position. The detection of a verb-object collocation in a passive sentence is thus triggered by the insertion of the empty constituent in direct object position. The collocation identification procedure checks whether the antecedent of the (empty) direct object and the verb constitute a (verb-object) collocation. In the example *The decision was made*, the noun *decision* of the collocation *to make a decision* precedes the verb.

Another transformation that can affect some collocation types is pronominalization. In such cases, it is important to identify the antecedent of the pronoun which can be found either in the same sentence or in the context. The example cited below illustrates a sentence where the pronoun *it* refers to the noun *money*. Since the pronoun is the subject of the passive form *would be well spent*, it is interpreted as direct object of the verb and therefore stands for an occurrence of the collocation *to spend money*:

...though where the money would come from, and how to ensure that it would be well spent, is unclear.

To handle them, the identification procedure sketched above must be slightly modified so that not only the attachment of a lexical item triggers the identification process, but also the attachment of the trace of a preposed lexical item. In such a case, the search will consider the antecedent of the trace. This shows, again, that the main advantage provided by a syntactic parser in such a task is its ability to identify collocations even when complex grammatical processes disturb the canonical order of constituents.

4 Setup for the shared task

In this section, we are going to present the experiment that was performed for French in the framework of the open track of the shared task on automatic identification of VMWEs and the modifications that were made to our parser in order to fulfill this task. Verbal MWEs include idioms (*let the cat out of the bag*), light verb constructions (*make a decision*), verb-particle constructions (*give up*)[1], and inherently reflexive verbs (*se taire, s'appuyer* 'to shut up', 'to rely on' in French).

4.1 Implementation

As the Fips parser already includes a collocation identification module and produces full syntactic trees for the constituents of the sentence, including the verbal constructions, our participation to the Shared Task consisted essentially in developing a transformation code between the PARSEME and Fips input - output formats. There were three kinds of transformation needed: (i) the reconstitution of the raw text from the tokenized one that was already provided (ii) the alignement of the provided tokens with the tokens generated by Fips and (iii) the copy of the Fips detected VMWE to the tokenized parsemetsv file, i.e. the annotation of the identified VMWEs.

4.1.1 Raw text

The Fips parser requires raw text input. This led us to develop a pre-processor that reconstructs the original text from the tokenized data provided for the shared task. This development was rather easy for French as the file included as a comment the original text for each given sentence. For the other languages, the pre-processor consisted in concatenating the tokens, taking into account the *ns* field indicating the presence or absence of a space character.

4.1.2 Tokens alignment

The shared task evaluation measures being token-based, for understandable evaluation reasons, the systems were asked to produce the results using strictly the same tokenization as those given in the data sets. In general, the parsemetsv and the Fips tokenization of words are identical but in numerous cases they differ. The trend in parsemetsv tokenization is to consider two words separated

[1] Verb-particle constructions don't exist in French, but they exist in German and English, languages for which we originally intended to participate.

by a space as two different tokens. On the other hand, the Fips tokenization procedure is based on linguistic criteria, i.e. a token is a significant lexical unit. Thus, Fips groups together two or more words if they form a complex lexical unit, for instance the French compound nouns *pomme de terre* ("potato"), the German preposition *je nach* ("according to") or complex fixed adverbial phrases such as *by and large*. On the other hand, Fips may treat single words as multiple tokens. For instance, the German compounds are decomposed, so that *Medaillengewinner* ("medal winner") will be presented as two tokens (*Medaillen* and *Gewinner*). The parsemetsv format exhibits some special treatment for some tokens, e.g. the contracted determiner *du* ("of the") in French that generates three lines of data or for the treatment of the hyphen.

What appeared at first glance like a first year Computer Science student assignment turn out to be a little bit more complicated.

4.1.3 VMWEs annotation

The Fips parser can produce several output formats: syntactic tree, tagger, XML/TEI, etc[2]. We chose the Fips tagger output developed for the SwissAdmin project (Scherrer et al., 2014) because it gives all the necessary information for the VMWE annotation and, like in pasemetsv, it outputs one token per line. In short, each (Fips) token is displayed on one line, divided in six columns: the token, the Universal POS tag, the richer Fips tag, the lemma, the grammatical function / valency (if any), the collocation (if any)[3]. The annotation of VMWEs is processed sentence by sentence and takes place as follows: the Fips output (aligned with the parsemetsv data file) is sequentially traversed line by line. For each verb token, the following tests are performed (in the following priority order). Note that in every case the annotations take place in the parsemetsv (aligned) data file:

- if the verb is reflexive, it is flagged; the Fips output is then traversed backward and the first encountered reflexive pronoun is flagged;

- if the verb is a light verb and the grammatical function displays a direct object, it is flagged; the Fips output is then traversed forward until the direct object is encountered; if the direct object is not encountered, a backward traversal is performed (in order to deal with the passive forms);

- if the verb is impersonal, the verb is flagged; the algorithm looks for the subject in order to annotate it;

- if the verb is part of a verbal collocation, it is flagged as OTH (OTHER) and a treatment similar to the one for the light verb is performed in order to annotate the complement(s).

5 Evaluation and results

Evaluation metrics are precision, recall and F1, both strict (per VMWE) and fuzzy (per token, i.e. taking partial matches into account). The token-based F1 takes into account:

- discontinuities (*take something into account*);

- overlapping (*take a walk and then a long shower*);

- embeddings both at the syntactic level (*take the fact that I didn't give up into account*) and at the level of lexicalized components (*let the cat out of the bag*).

However, VMWE categories (e.g., LVC, ID, IReflV, VPC) were ignored by the evaluation metrics.

We measured the best F1 score from all possible matches between the set of MWE token ranks in the gold and system sentences by looking at all possible ways of matching MWEs in both sets. In the evaluation per MWE, our system achieved 0.4815 precision with a recall of 0.4680 and F-measure of 0.4746. In the evaluation per token, our system achieved 0.5865 precision with a recall of 0.5108 and F-measure of 0.5461.

6 Conclusion

The good performance achieved by the Fips system confirms that deep syntactic information helps to identify MWEs and especially VMWEs. Although the VMWE annotation would be more accurate if it was based on the syntactic tree, the "flat" rich tagger output chosen for the alignment ease with the required parsemetsv tokenization was a good solution. An enhancement to this output would be to implement a token identification scheme so as to establish explicit links between the verbs and their arguments (instead of sequentially traverse the sentence and rely on the orthographic form of the word).

[2] The Fips parsing service is available at http://latlapps.unige.ch/Parser

[3] See Scherrer et al. (2014) for more details and examples.

References

Iñaki Alegria, Olatz Ansa, Xabier Artola, Nerea Ezeiza, Koldo Gojenola, and Ruben Urizar. 2004. Representation and treatment of multiword expressions in basque. In Tanaka Takaaki, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors, *Second acl workshop on multiword expressions: integrating processing*. Association for Computational Linguistics.

Caroline Brun. 1998. Terminology finite-state preprocessing for computational lfg. In *Proceedings of COLING 1998*, page 196–200.

Matthieu Constant and Joakim Nivre. 2016. A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 161–171, Berlin, Germany.

Ulrich Heid. 1994. On ways words work together – topics in lexical combinatorics. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

Alexis Nasr, Carlos Ramisch, José Deulofeu, and André Valli. 2015. Joint dependency parsing and multiword expression tokenization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1116–1126, Beijing, China.

Yves Scherrer, Luka Nerima, Lorenza Russo, Maria Ivanova, and Eric Wehrli. 2014. Swissadmin: A multilingual tagged parallel corpus of press releases. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Violeta Seretan. 2011. *Syntax-Based Collocation Extraction*. Springer.

Aline Villavicencio, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

Eric Wehrli and Luka Nerima. 2013. Sentence analysis and collocation identification. In *Proceedings of the Workshop on Multiword Units in Machine Translation and Translation Technology, MT Summit XIV*, pages 12–17, Nice, France.

Eric Wehrli and Luka Nerima. 2015. The fips multilingual parser. In Nuria Gala, Reinhard Rapp, and G. Bel, editors, *Festschrift in honour of Michael Zock*. Springer.

Eric Wehrli. 2007. Fips, a deep linguistic multilingual parser. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Parsing*, pages 120–127, Prague, Czech Republic.

Eric Wehrli. 2014. The relevance of collocations for parsing. In *Proceedings of the 10th Workshop on Multiword Expressions, MWE@EACL 2014*, pages 26–32, Gothenburg, Sweden.

Yi Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proceedings of LREC-2006*, Genoa, Italy.