A Recursive Treatment of Collocations

Luka Nerima, Eric Wehrli, Violeta Seretan
LATL – Département de Linguistique
Université de Genève
2, rue de Candolle, CH-1211 Genève 4, Switzerland
E-mail: {luka.nerima, eric.wehrli, violeta.seretan}@unige.ch

Abstract

This article discusses the treatment of collocations in the context of a long-term project on the development of multilingual NLP tools. Besides "classical" two-word collocations, we will focus on the case of complex collocations (3 words or more) for which a recursive design is presented in the form of collocation of collocations. Although comparatively less numerous than two-word collocations, the complex collocations pose important challenges for NLP. The article discusses how these collocations are retrieved from corpora, inserted and stored in a lexical database, how the parser uses such knowledge and what are the advantages offered by a recursive approach to complex collocations.

1. Introduction

Among the many issues concerning collocations in NLP, this paper considers the recursive nature of collocations. One finds, along with the classical examples of collocations consisting of two terms - *foil-attempt, experience-problem, action-plan, lever-séance (to adjourn the meeting)*, *tenir-compte (to take into account)*, *grièvement-blessé, (seriously injured)*, *gravement-malade (seriously ill)* - expressions containing 3 or more terms. Here are some examples: *weapons of mass destruction*, *unmarked police car*, *agent de la force publique (member of the police force)*, *tomber en panne sèche (to run out of gas)*, etc. We henceforth refer to such expressions as to complex collocations.

The issues we address in this paper are (i) the insertion and representation of complex collocations in the lexical database and (ii) the identification and processing of such expressions by a syntactic parser. The approach we propose is based on the idea that the structure of a collocation is recursive in nature. A collocation is composed of two phraseological units which can be either single words or poly-lexical units. In other words, we consider the possibility of collocations of collocations.

Similar positions were advocated in other articles, for instance in (Heid, 1994; Tutin, 2008). However, to the best of our knowledge they did not lead to an implementation in a NLP system. In this paper, we present the manner in which complex collocations are taken into account in our NLP environment, in which they undergo a full processing cycle (from the identification in source corpora and their storage into a lexical database, to their use in a syntactic parser and a machine translation system).

The article is structured as follows. In Section 2, we address some theoretical issues related to the notion of complex (recursive) collocations, then give a brief overview of the existing practical approaches, and finally present the motivation for a recursive treatment. In Section 3, we describe our methods of acquiring complex collocations from corpora, discuss their representation in the lexicon, and present our tools for insertion in our lexical database. We show the advantages of adopting a recursive structure for these collocations. In Section 4 we show how the treatment of complex collocations takes place in our parsing system, and we conclude in Section 5 with some general remarks.

2. Motivation

The notion of "collocation" is difficult to define precisely and tends to vary from one author to another (see for instance Seretan, 2008 for an overview of definitions). In spite of that, the need and importance of the role played by collocations in many practical applications is widely recognized in the NLP community. As for us we only consider collocations which form a syntactic constituent, i.e. collocations whose components are linked by a syntactic relation.

Although several authors have compiled a list of possible collocation configurations (for instance, Hausman, 1989) we do not limit ourselves to a closed list. In addition to the syntactic configuration, the collocations obey certain constraints on their specific form (e.g., plural collocation, possessive complement, bare noun complement).

A syntactic approach to collocations naturally leads us to consider collocation as intrinsically recursive: the components of a collocation can be collocations themselves. For instance, the collocation Noun + Noun *mass destruction* can combine with the noun *weapon*. As the syntactic head of *mass destruction* is a noun, it combines with *weapon* and leads to the complex collocation *weapons of mass destruction* of type Noun+Prep+Noun.

From a theoretical point of view, it is taken for granted that collocations consist of "two or more words" (Sinclair, 1991). It is also well known that collocations may combine to yield more complex collocations of (virtually) unrestricted length. In fact, researchers like Heid (1994) have long since remarked the recursive nature of collocations. Yet, the practical work deals almost exclusively with binary collocations, made up of only two words.

One reason for this situation is the scarcity (or rather the absence) of association measures of higher arity, which apply to candidates longer than two items. Most measures based on hypothesis testing and on contingency tables only deal with association between two elements. Some efforts have been made to generalize measures like MI (mutual information) to *n* elements, with more or less success (see, for instance, Villada Moirón, 2005) . These measures remained, however, unpopular.

Another challenge posed by complex collocations is the combinatorial explosion when considering all possible word combinations of length *n* as candidates. When faced with the huge number of possible combinations and the increase of the search space, most of the existing approaches adopt the strategy of considering only sequences of consecutive words (*n*-grams), and filtering out the functional words because they are very frequent. Such accounts fail (e.g. Dias, 1991), however, to model the syntactic flexibility of collocations, which is the key feature of this subclass of multi-word expressions. At the same time, they impose too serious restrictions on the syntactic configuration by eliminating certain functional words which are essential, like prepositions.

Due to these reasons, complex collocations remain a rather unexplored area of research and they are usually ignored in the existing practical work. There are nonetheless important motivations for providing a proper account for this kind of collocations in a NLP environment.

The main motivation, behind the purely theoretical interest, is that existing technologies of lexical acquisition also provide, among the resulting binary associations, incomplete subparts of complex collocations. Pairs of words are found that make no sense in isolation, as they only represent fragments of complex collocations. This problem was more acutely perceived in the related field of terminology, where a combination like *soft contact lens* cannot be decomposed into fragments like *soft lens* or *soft contact*, but must be recovered in its entirety (Frantzi et al., 2000).

Another motivation is that, even if some binary combinations cannot be considered as fragments because they also occur in isolation, there is a strong preference for them to be used in conjunction with other words in order to acquire a syntactically complete status. This is for instance the case of the binary collocation *stand – in – contrast*. In many contexts, the noun *contrast* requires a modifier that would complete the expression; in fact, the modifier which is the most used is *stark*. Therefore, a lexicographer might find it useful to record the whole expression *stand in stark contrast*, rather than (or in addition to) the individual subparts alone, *stand in contrast* and *stark contrast*.

Another example of a nested collocation is *mass destruction*. The lexicographer might decide that, in addition to this binary combination, it would also be useful to store larger collocations of which this is usually part, and so on; for instance:

(1) weapons of mass destruction
proliferation of weapons of mass destruction
treaty on the non-proliferation of weapons of mass destruction

Summing up, the interest in complex collocations is both theoretically and lexicographically motivated; this is an issue which is given particular attention in our NLP work.

3. A recursive approach

As suggested in Section 1, the notion of collocation from co-occurrence of words can be extended to co-occurrence of collocations. Thus, by exploiting the recursive nature of collocations, we can apply the same extraction methodology as in the case of two-word collocations, in order to obtain complex collocations of unrestricted length. Thus, we rely on the recursive nature of collocations in order to define a convenient lexical representation for these complex collocations in our lexical database.

3.1. Acquisition of candidates from corpora

In our syntactically-based extraction of binary collocations (Seretan, 2008), the search space for candidates in a corpus is defined as the set of lexical item pairs in a direct syntactic relationship (like verb-object, adjective-noun, adverb-adjective etc). An association measure is then applied on these candidates in order to obtain a ranking according to their collocational strength.

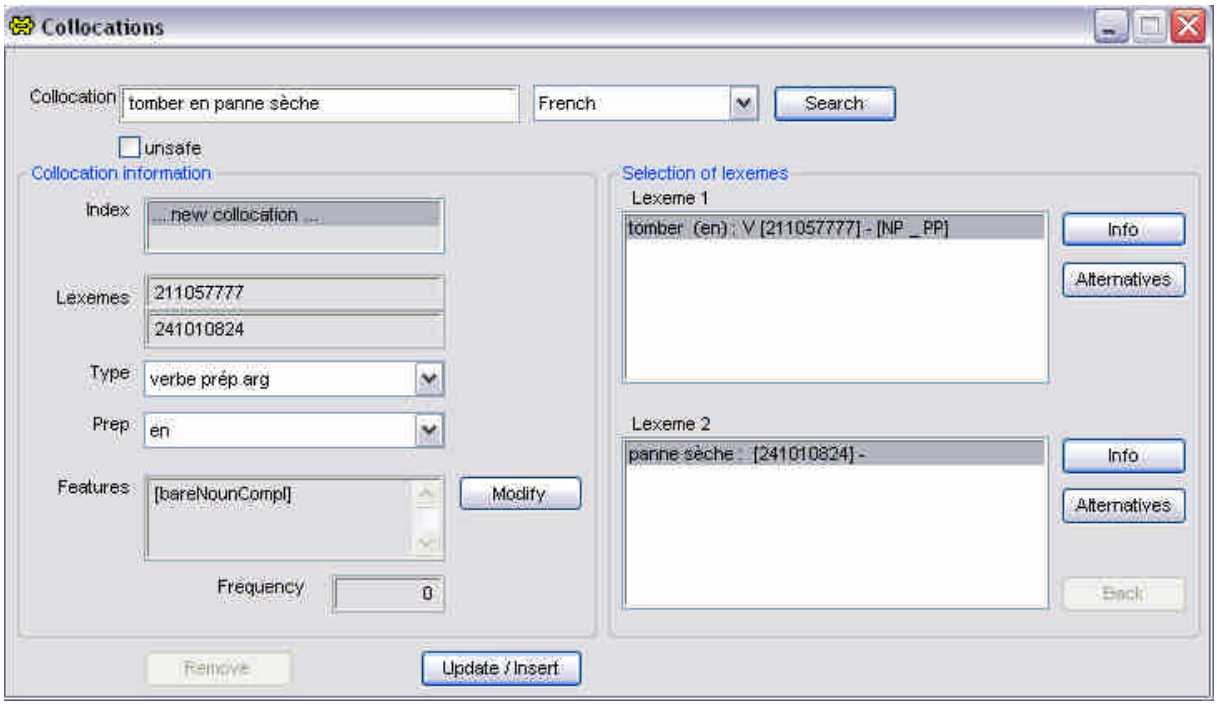
Since the syntactic configurations which are appropriate for complex collocations are much more numerous than for binary collocations and cannot be known in advance, it was impossible to devise a similar extraction. Instead, we found a different solution, in which we identify complex collocations by relying on their recursive nature, i.e., by viewing complex collocations as co-occurrence of previously extracted binary collocations, rather than co-occurrence of single words.

Cooccurrence of two binary collocations means, more exactly, that they combine syntactically by sharing a common term in the same sentence. For instance, *contrast* is shared by both of the binary collocations *stand in contrast* and *stark contrast* previously identified in the sentence (2). Their combination therefore yields the complex collocation *stand in stark contrast*.

(2) The apparent remoteness and peacefulness of the area *stand in stark contrast* to the bustling city.

This is the main means we use for extracting recursive collocation candidates. In addition, complex collocations are also obtained in our framework as a side-effect of our standard extraction procedure. More precisely, when a collocation which is present in the parser's lexicon is identified in the source text by Fips, it is treated by the extractor as a single unit, and is further considered as a term of a new (binary) collocation.

For instance, if *stark contrast* is added to our lexical database, the parser will recognize it in future analyses, i.e., when it processes a sentence like the one in (2). The extractor will then consider the whole phrase, *stark*



contrast, as the argument of the verb *stand*. The candidate *stand – in – stark contrast* will therefore be proposed as a binary collocation candidate of Verb+Preposition+Noun type.

3.2. Insertion in the lexicon

Insertion of collocations in our lexical database is performed under the supervision of a lexicographer. Automatic tools for terminology extraction (including extraction of complex collocations) like those described in (Seretan et al., 2004) applied to large corpora provide good candidates, enriched with statistical information and with contexts of usage. The lexicographer selects among these candidates the relevant collocations and inserts them in the lexicon.

As we expect that a large number of collocations will be entered in our lexicons, probably several thousands per language, the usability of the user interface is a crucial point. When the lexicographer types in a new collocation, a full syntactic analysis is performed by the parser. As a result of the parsing, the interface proposes:

- the components of the collocation (words or collocations);
- the syntactic configuration (Adj + Noun, Verb + Object, Noun + Prep + Noun, etc.);
- the preposition used, if any (e.g., *of* in *hall **of** fame*);
- the morphosyntactic features which constrain the collocation form.

It is then up to the lexicographer to validate or to modify the proposed parameters. The accuracy of these parameters depends on the performance of the parser for each specific language. Our experience has showed that both for French and English the system does well in more than 90% cases.

To illustrate the process of analysis of recursive collocations, we provide an example in Figure 1. It shows the graphical user interface after the system has parsed the collocation *tomber en panne sèche* (*to run out of gas*), which the user intends to insert in the lexical database. As the first component of the collocation, the interface proposes the appropriate lexeme of the verb *tomber*, that is, the one with a PP argument and the preposition *en*. As the second component, it proposes the collocation *panne sèche*. The parser also determines the syntactic configuration Verb + Prep + Arg (field "Type"), as well as the constraint "BareNounCompl".

In case of an inappropriate suggestion, the user can correct the choices of the parser. Most likely, the user may want to choose a different reading for a collocation component. The system displays all the alternative readings available in the lexical database, from which the user can select the correct one manually.

3.3. Advantages

A recursive approach to collocations shows several advantages:

- Generality: the same descriptive structure may be used. The syntactic configuration of the collocation applies whether components are words or collocations. The same parsing algorithm can be used to process collocations of collocations as well as collocations of words.
- Extensibility: recursion allows taking into account collocations of arbitrary length.

- Reusability: each embedded collocation has its own description of syntactical configuration and morphosyntactic features, hence it is not necessary to repeat it in the framework of the global collocation. The complex collocation inherits the descriptions of the embedded collocations.

- Uniformity of the underlying linguistic theory: the composition of the partial syntactic analysis of the collocations leads to the same syntactic analysis as the one the parser would produce without the knowledge of the collocations.

4. Treatment of complex collocations

In this section we briefly describe the method used in the Fips parser (Wehrli, 2007; Wehrli and Nerima 2009) to handle collocations. Collocation identification is particularly important for instance when the parser is used in the context of a machine translation system. It is indeed well known that collocations, in general, cannot be translated literally (*panne sèche ≠ dry breakdown**, *heavy smoker ≠ lourd fumeur**, *caresser l'espoir ≠ caress the hope)*.

One of the main difficulties in collocation identification comes from the fact that the two terms (simple or complex) of the expression are not necessarily immediately adjacent to each other. In addition to this, in the case of collocations of type Verb + Object, the object is able to undergo syntactic transformations (e.g., relativization, passivization, topicalization). In order to handle the flexibility of collocations, a "deep" linguistic analysis must be performed. For instance, in order to identify the collocation *break record* in Example 3

(3) the record that John broke was owned by Paul

the parser must be able (i) to recognize the presence of a relative sentence, (ii) to determine the syntactic role of the relative pronoun *that* in relation to the verb of the relative sentence (direct object), and (iii) to identify the antecedent of the relative pronoun (*record*). For the sentence fragment in Example 3, the Fips parser generates the following syntactic structure:

[$_{TP}$ [$_{DP}$ the [$_{NP}$ record-i [$_{CP}$ that-i [$_{TP}$ [$_{DP}$ John] [$_{VP}$ broke [$_{DP}$ e-i]]] was...]

Notice that in this analysis the noun *record* is coindexed with the relative pronoun *that*, which is in turn coindexed with the empty direct object of the verb *broke*. Given this antecedent-trace chain, it is relatively easy for the system to identify the Verb + Object collocation *break-record*.

It is on the basis of such structures that the parser performs the identification of collocations. In the case of a Verb + Object collocation, it considers the verbal lexical head *broke* and the semantic head of the direct object or of its antecedent (i.e. the head of the chain) if the latter is empty, as in our example. Thanks to this technique, the distance between the two terms of the collocation represents no particular problem (Goldman and al., 2003).

We now consider the parsing of recursive collocations, i.e., collocations in which one of the terms is a collocation itself. Consider for instance the examples in (4) below:

(4)a. la voiture tombera probablement en panne d'essence
"the car will probably run out of gas"

b. weapons of mass destruction

c. natural language processing

d. he broke a world record

In the French sentence (4)a, *panne d'essence* (litt: *breakdown of gas*, *out of gas*) is a collocation of type Noun + Prep + Noun, which combines with the verb *tomber* (litt. *fall*) to form a larger collocation of type Verb+PrepObject *tomber en panne d'essence* (*to run out of gas*). Given the strict left to right order of processing that Fips assumes, it will first find the *tomber en panne* (*to break down*) when processing the word *panne*. When it reads the word *essence* (*gas*), the presence of the (lexical) feature "part of collocation" triggers the collocation identification procedure, which examines whether a collocate can be found among its governors or its modifiers. In our example, the word *panne* is found, and a lookup in the collocation database validates the collocation *panne d'essence* (*out of gas*). Given the fact that the collocation *panne d'essence* bears the feature "part of collocation" too, the collocation identification procedure is triggered again searching for a collocate of that collocation. The search succeeds with the verb *tomber*, and the collocation *tomber en panne d'essence* (*run out of gas*) is identified.

The identification of the complex collocations in the next three examples of (4) is achieved in a similar way. For instance, in the example (c), the parser will read *natural* and *language*. Both bear the feature "part of collocation" and the collocation procedure will validate *natural language* as an adjective-noun collocation, which itself bears the feature "part of collocation". Reading the next word, *processing*, the identification procedure will consider and validate *natural language processing*.

5. Conclusion

Although comparatively less numerous than two-word collocations, complex collocations consisting of three or more words pose important challenges for NLP, both with respect to their lexical representation and their treatment. Currently, our lexical database contains a few hundreds recursive collocations for French and English, and we are pursuing efforts to increase the coverage.

In this article, we presented our method of handling complex collocations and we showed that there are several advantages in considering them as recursive structure: economy of description, since the description of the embedded collocations are inherited by the embedding collocations; efficiency in entering complex collocations in the lexical database, since the

lexicographer uses the same interface as the one for binary collocations; economy in processing, since the parser algorithm does not need significant modifications in order to be able to identify them. Properly identifying complex collocations is the foremost condition for processing them efficiently.

6. Acknowledgements

This research was partially supported by the Swiss National Fund, grant no 100015-117944.

7. References

Benson, M. (1990). Collocations and general-purpose dictionaries. International Journal of Lexicography, 3:1, 23-35.

Dias, G. 2003. Multiword unit hybrid extraction. In *Proceedings of the ACL Workshop on Multiword Expressions*, pages 41–48, Sapporo, Japan.

Goldman, J-P., Nerima, L. and Wehrli, E. (2003). Collocation Extraction Using a Syntactic Parser. In Proceedings of the ACL Workshop on Collocations, Toulouse, France, 61-66.

Frantzi, K. T., Ananiadou, S. and Mima, H. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 2(3):115–130.

Hausmann, F. J. (1989). Le dictionnaire de collocations. In Hausmann, F.J. et al. (éds), Wörterbücher: ein internationales Handbuch zur Lexicographie. Dictionaries, Dictionnaires, Berlin, de Gruyter, 1010-1019.

Heid U. (1994). On ways words work together – research topics in lexical combinatorics. In Proceedings of the VIth Euralex International Conference Congress on Lexicography (EURALEX'94), Amsterdam, The Netherlands, 226-257.

Seretan, V., Nerima, L. and Wehrli, E. (2004). A Tool for Multi-Word Collocation Extraction and Visualization in Multilingual Corpora. In Proceedings of the Eleventh EURALEX International Congress (EURALEX 2004), Lorient, France, 755-766.

Seretan, V. (2008) Collocation Extraction Based on Syntactic Parsing. Thèse de doctorat, Université de Genève.

Tutin, A. (2008). For an extended definition of lexical collocations. In Proceedings of Euralex, Barcelone 15-19 juillet 2008, Université Pompeu Fabra.

Villada Moirón, M. B. 2005. *Data-driven identification of fixed expressions and their modifiability*. Ph.D. thesis, University of Groningen.

Wehrli, E. (2007). A "Deep" Linguistic Multilingual Parser. In Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing, Prague, Czech Republic, 120-127.

Wehrli, E. and Nerima, L. (2009). L'analyseur syntaxique Fips. In IWPT'09 ATALA Workshop: What French parsing system?, Paris, France, October. Association pour le traitement automatique des langues.

Zinmeister, H. and Heid, U. (2003). Significant triples: Adjective+Noun+Verb combinations. In Proceedings of the 7th Conference on Computational Lexicography and Text Research (Complex 2003), Budapest, Hungary.