# Discovering linguistic patterns using sequence mining

Nicolas Béchet, Peggy Cellier, Thierry Charnois, Bruno Crémilleux

# Discovering Linguistic Patterns using Sequence Mining

Nicolas Béchet[1], Peggy Cellier[2], Thierry Charnois[1], and Bruno Cremilleux[1]

[1] GREYC Université de Caen Basse-Normandie
Campus II science 3
14032 Caen CEDEX, France.
{nicolas.bechet, thierry.charnois, bruno.cremilleux}@unicaen.fr
[2] INSA Rennes/IRISA,
Campus de Beaulieu
35042 Rennes cedex, France
peggy.cellier@irisa.fr

**Abstract.** In this paper, we present a method based on data mining techniques to automatically discover linguistic patterns matching appositive qualifying phrases. We develop an algorithm mining sequential patterns made of itemsets with gap and linguistic constraints. The itemsets allow several kinds of information to be associated with one term. The advantage is the extraction of linguistic patterns with more expressiveness than the usual sequential patterns. In addition, the constraints enable to automatically prune irrelevant patterns. In order to manage the set of generated patterns, we propose a solution based on a partial ordering. A human user can thus easily validate them as relevant linguistic patterns. We illustrate the efficiency of our approach over two corpora coming from a newspaper.

## 1 Introduction

Due to the explosion of available textual data, the need for efficient processing of texts has become crucial for many applications; for instance, extraction of biological knowledge from biomedical texts, monitoring opinion from newspapers or forums. Natural Language Processing (NLP), and Information Extraction (IE) in particular, aim to provide accurate parsing to extract specific knowledge such as named entities (e.g., gene, person, company) and relationships between the recognized entities (e.g., gene-gene interactions). A common feature of the information extraction methods is the need for linguistic resources (grammars or linguistic rules). This paper deals with this problem and proposes a method for automatically discovering linguistic patterns.

Indeed, NLP approaches apply rules such as regular expressions for surface searching [10] or syntactic patterns [9]. However, these rules are handcrafted and thus those methods are time consuming and very often devoted to a specific corpus [11]. In contrast, machine learning based methods such as support vector machines or conditional random fields [13], are less time consuming than NLP methods. Although they provide good results, they still need many features. Moreover, their outcomes are not really understandable by a user, nor they can be used as linguistic patterns in NLP systems (because the produced models are numerical). Furthermore, the annotation process of training corpora requires a substantial investment of time, and cannot be reused in other

domains [11] (annotation of new corpora in new domains requires to repeat this time consuming work).

A promising avenue is the trade-off coming from the cross-fertilization of information extraction and machine learning techniques which aims at automatically learning linguistic resources such as lexicons or patterns [14]. Most of these symbolic approaches are supervised. RAPIER, a well-known system based on inductive logic programming, learns information extraction rules [3] but uses annotated corpora difficult to acquire as previously explained. A few of unsupervised approaches have been designed: one of these earliest works presents a method to acquire linguistic patterns from plain texts but it needs a syntactic parsing [16]. Therefore, the quality of learned patterns stems from the syntactic process results. New works take advantage of an hybridization of data mining and NLP techniques. An advantage of data mining techniques is to enable the discovery of implicit, previously unknown, and potentially useful information from data [8]. For instance, *Cellier et al.* [4] aim at discovering linguistic rules to extract relationships between named entities in new corpora. That approach is not supervised and does not need syntactic parsing nor external resources except the training corpus. It relies on extraction of frequent sequential patterns where a sequence is a list of literals called *items*, and an item is a word (or its lemma) within textual data. A well-known limitation of data mining techniques is the large set of discovered patterns. It needs to be filtered or summarized in order to return only relevant patterns. In the sequel, we present how we address this problem thanks to constraints and partial order.

The contribution of this paper is twofold. First, we improve works such as [4] by being able to handle sequences of *itemsets* instead of *single-items*. That means that a word can be represented by a set of features conveying several pieces of information (e.g., words, lemma) and not only a single information. Thus the extracted patterns combine different levels of abstraction (e.g., words, lemma, part of speech tags) and express information according to different levels of genericity, for example $\langle (champion\ NOUN)\ (of\ PRP)\ (the\ DET)\ (world\ NOUN) \rangle$ and $\langle (champion\ NOUN)\ (PRP)\ (DET)\ (NOUN) \rangle$ (see Section 2.3 for details). We have developed an algorithm for discovering such sequential patterns under constraints. Indeed, constraints enable to add user knowledge into the discovery process in order to give prominence to the most significant patterns. Secondly, we tackle the problem of pattern selection by proposing a tool allowing a user to easily navigate within the pattern space and validate sequential patterns as linguistic patterns. The navigation and validation take advantage of the partial order between patterns.

We apply our approach on learning linguistic patterns for discovering phrases denoting judgment or sentiment in French texts, and more generally qualification as given in Table 1 and called appositive qualifying phrases. It is important to note that our approach is not dedicated to a specific kind of linguistic patterns nor a specific language, but it can easily be adapted to other information extraction applications (e.g., relationships between named entities) or other languages. Indeed, our method is based on sequence mining techniques which are not language-dependent.

In the remaining of the paper, Section 2 introduces the method to extract sequential patterns and validate them. Section 3 presents and discusses experiments about appositive qualifying phrases.

| sid | Sequence |
|---|---|
| 1 | $\langle(hommes\ homme\ NOUN)\ (de\ PRP)\ (culture\ NOUN)\rangle$ |
| 2 | $\langle(femmes\ femme\ NOUN)\ (de\ PRP)\ (conviction\ NOUN)\rangle$ |
| 3 | $\langle(charismatique\ ADJ)\ (et\ KON)\ (ambitieux\ ADJ)\rangle$ |
| 4 | $\langle(réputé\ réputer\ VER\ pper)\ (pour\ PRP)\ (sa\ son\ DET\ POS)\ (cruauté\ cruauté\ NOUN)\rangle$ |

**Table 1.** Excerpt of sequential database for French texts: "*homme de culture*" (*intellectual man*), "*femme de conviction*" (*woman of conviction*), "*charismatique et ambitieux*" (*charismatic and ambitious*), "*réputé pour sa cruauté*" (*famous for his violence*).

## 2 Extraction of Linguistic Patterns

Our approach is a two step method. First we extract sequential patterns thanks to our sequence mining algorithm. Secondly, we organize them in a data structure according to a partial order so that a linguist expert can easily validate extracted sequential patterns as linguistic patterns. More precisely, Section 2.1 introduces background knowledge about sequential patterns. Then we explain how constraints are at the core of the process to produce relevant candidate linguistic patterns (cf. Section 2.2). Finally, Section 2.3 presents the validation step.

### 2.1 Sequential Pattern Mining

Sequential pattern mining is a well-known data mining technique introduced in [1] to find regularities in a sequence database. There is a lot of algorithms to extract sequential patterns [19, 21, 20, 22]. That point is discussed in Section 2.2.

In sequential pattern mining, an *itemset* $I$ is a set of literals called *items*, denoted by $I = (i_1 \dots i_n)$. For example, $(homme\ NOUN)$ is an itemset with two items: $homme$ and $NOUN$. A *sequence* $S$ is an ordered list of itemsets, denoted by $s = \langle I_1 \dots I_m\rangle$. For instance, $\langle(hommes\ \ homme\ \ NOUN)(de\ \ PRP)(culture\ \ NOUN)\rangle$ (coming from Table 1) is a sequence of three itemsets. A sequence $S_1 = \langle I_1 \dots I_n\rangle$ is *included* in a sequence $S_2 = \langle I'_1 \dots I'_m\rangle$ if there exist integers $1 \leq j_1 < ... < j_n \leq m$ such that $I_1 \subseteq I'_{j_1}$,..., $I_n \subseteq I'_{j_n}$. The sequence $S_1$ is called a *subsequence* of $S_2$, and we note $S_1 \preceq S_2$. For example, $\langle(NOUN)(de\ PRP)\rangle$ is included in $\langle(hommes\ homme\ NOUN)(de\ PRP)(culture\ NOUN)\rangle$. A sequence database $SDB$ is a set of tuples $(sid, S)$, where $sid$ is a sequence identifier and $S$ a sequence. For instance, Table 1 depicts a sequence database of four sequences. A tuple $(sid, S)$ *contains* a sequence $S_1$, if $S_1 \preceq S$. The *support* of a sequence $S_1$ in a sequence database $SDB$, denoted $sup(S_1)$, is the number of tuples in the database containing $S_1$[3]. For example, in Table 1 $sup(\langle(NOUN)(de\ PRP)\rangle) = 2$, since Sequences 1 and 2 contain $\langle(NOUN)(de\ PRP)\rangle$. A *frequent sequential pattern* is a sequence such that its support is greater or equal to a given support threshold $minsup$.

The set of frequent sequential patterns can be very large. Condensed representations, such as closed sequential patterns [21], have been proposed in order to eliminate redundancy without loss of information. A frequent sequential pattern $S$ is closed if

[3] Note that the *relative support* is also used: $sup(S_1) = \frac{|\{(sid,S)\ |\ (sid,S) \in SDB \wedge (S_1 \preceq S)\}|}{|SDB|}$.

there is no other frequent sequential pattern $S'$ such that $S \preceq S'$ and $sup(S) = sup(S')$. For instance, with $minsup = 2$, the sequential pattern $\langle(NOUN)(NOUN)\rangle$ from Table 1 is not closed whereas $\langle(NOUN)(de\ PREP)(NOUN)\rangle$ is closed.

The constraint-based pattern paradigm [6] brings useful techniques to express the user's interest in order to focus on the most promising patterns. A very well-used constraint is the frequency. A sequence $S$ is frequent if and only if $sup(S) \geq minsup$ where $minsup$ is a threshold given by a user. However, it is possible to define many other useful constraints such as the gap constraint. A gap is a sequence of itemsets which may be skipped between two itemsets of a sequence $S$. $g(M, N)$ represents a gap whose size is within the range $[M, N]$ where $M$ and $N$ are integers. The range $[M, N]$ is called a *gap-constraint*. A sequential pattern satisfying the gap-constraint $[M, N]$ is denoted by $P_{[M,N]}$. It means there is a gap $g(M, N)$ between every two neighbor itemsets of $P_{[M,N]}$. For instance, in Table 1, $P_{[0,2]} = \langle(PRP)(NOUN)\rangle$ and $P_{[1,2]} = \langle(PRP)(NOUN)\rangle$ are two patterns with gap constraints. Indeed, $P_{[0,2]}$ matches three sequences (1, 2 and 4) whereas $P_{[1,2]}$ matches only Sequence 4.

### 2.2 Algorithm to Extract Sequential Patterns

We present in this section our algorithm mining the closed sequential patterns of itemsets under constraints. There are already in the literature many algorithms to extract sequential patterns (e.g. `GSP` [19], `SPADE` [22], `PrefixSpan` [15]) or closed sequential patterns (e.g. `CloSpan` [21], `BIDE` [20]). But, to the best of our knowledge, there is no algorithm mining closed sequential patterns made of itemsets under constraints able to take into account the field of knowledge. In this paper, we address this open issue by proposing an algorithm mining sequential patterns made of itemsets under various constraints.

Adding constraints to the sequential pattern mining process is not trivial. The combination of constraints and the closure must be properly managed [2] in order to get the correct condensed representations of patterns with respect to the constraints. That is why our algorithm considers the closure after applying constraints to provide the pattern condensed representation. More precisely, sequential patterns satisfying the frequency and gap constraints are firstly produced, then the closed patterns are computed. Details of the algorithm are not given in this article because it is out of the scope of the paper.

We introduce the *begin_with* constraint which is very useful on textual data. A sequential pattern $P$ satisfies the *begin_with* constraint if there is at least one sequence from SDB having its first itemset containing the first itemset of $P$. For instance, the sequential pattern $\langle(NOUN)\ (PRP)\ (culture\ \ NOUN)\rangle$ satisfies the $begin\_with$ constraint in SDB (cf. Table 1) because its first itemset $(NOUN)$ belongs to the first itemset of Sequence 1 $\langle(hommes\ homme\ NOUN)\ (de\ PRP)\ (culture\ NOUN)\rangle$. This constraint is precious to highlight appositive qualifying phrases. This one means that the appositive qualifying phrases has to appear at the beginning of a sequence. Moreover, we use a gap constraint of $g(0, 0)$ because appositive qualifying phrases are often made up of contiguous elements, which means that extracted patterns need to have contiguous itemsets according to the original sequences.

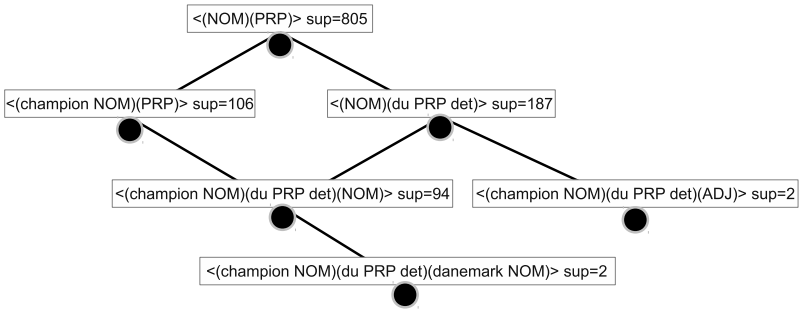


**Fig. 1.** Excerpt of the partial order on the patterns extracted from a corpus

### 2.3 Validation of Sequential Patterns as Linguistic Patterns

Constraints and closure reduce the set of extracted sequential patterns by pruning irrelevant patterns. Nevertheless, the number of extracted patterns can remain high. It is thus difficult for a human expert to validate them by hand as relevant linguistic patterns.

However, the set of extracted sequential patterns is partially ordered. Indeed, some patterns are more specific than others. For example, $\langle(champion\ \ NOUN)\ (du\ \ PRP\ \ det)\ (danemark\ \ NOUN)\rangle$ is more specific than $\langle(champion\ NOUN)\ (du\ PRP\ det)\ (NOUN)\rangle$. Thus the sentences matched by the pattern $\langle(champion\ NOUN)\ (du\ PRP\ det)\ (danemark\ NOUN)\rangle$ are also matched by the pattern $\langle(champion\ NOUN)\ (du\ PRP\ det)\ (NOUN)\rangle$. Therefore, when an expert selects a sequential pattern to promote it as a relevant linguistic pattern, she does not have to take care of more specific ones. We propose to take advantage of that partial order to organize the sequential patterns in a data structure, in order to help an expert to explore and select sequential patterns as linguistic patterns. The data structure is given in the form of a Hasse diagram [5]. Figure 1 shows an excerpt of a Hasse diagram for six sequential patterns extracted from one of our corpora. Nodes are sequential patterns, and edges between nodes represent the partial order relation.

The Hasse diagram can be very large. That is why we propose to use Camelis[4] [7], a tool which allows to navigate in partial orders. Figure 2 shows the Camelis interface. At the bottom part, the *navigation tree* displays the patterns. The partial order over the set of patterns is highlighted by the navigation tree. The navigation tree is not a tree structure but represents a partial order and a pattern can have several parents. It explains why in Figure 2 the pattern $\langle(champion\ NOUN)\ (du\ PRP\ det)\ (NOUN)\rangle$ appears twice in the navigation tree (this pattern has two parents). The number on the left of a pattern is the number of patterns which are more specific. For example, in Figure 2, 235 sequential patterns are more specific than the pattern $\langle(NOUN)\ (PRP)\rangle$. The support of $\langle(NOUN)\ (PRP)\rangle$ is 805 meaning that in the learning corpus 805 phrases contain a noun followed by a preposition.

[4] http://www.irisa.fr/LIS/ferre/camelis/index.html

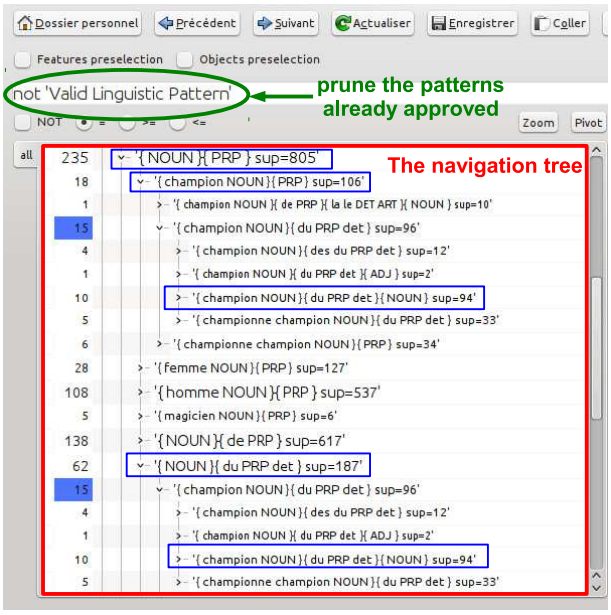


**Fig. 2.** Example of navigation from Camelis in order to validate linguistic patterns.

At the top, the query view displays the current query. In Figure 2, the query is "not Valid Linguistic Pattern", i.e. the displayed patterns are the patterns not already selected as relevant linguistic patterns. Indeed, when exploring the patterns the expert may add some information about the patterns. The two main advantages of the process are: first, it enables the user to easily navigate in the sequential pattern set in order to validate them and, secondly, it allows to prune patterns without interest (i.e. sequential patterns already selected as linguistic patterns or patterns identified as not linguistic patterns) and thus reduce the exploration space. If a pattern $P$ is selected as a relevant linguistic pattern, all more specific patterns than $P$ are filtered out, i.e. these patterns do not have to be explored because the phrases matched by them are also matched by $P$.

## 3 Application: The Appositive Qualifying Phrases

### 3.1 Appositive Qualifying Phrases

In the opinion analysis framework, one crucial task is the extraction of phrases expressing judgement or qualification (the distinction is out of our scope). In the remaining of the paper, we call that kind of phrases: *appositive qualifying phrases*. Those phrases check some syntactic criteria: they have a relative free position in the sentence ; they are bounded by punctuation ; they are compounded of contiguous words (see [12] for more linguistic details). Some examples are given below (in bold font):

**(1)** *Mais,* **en politicien expérimenté***, élu pour la première fois à la Knesset il y a trente-cinq ans, il a su résister aux roquettes de ses adversaires politiques.* (But, **as**

**a real politician**, elected for the first time at the Knesset 35 years ago, he managed to face his political opposant attacks.)

**(2)** **Ni trop sentimental**, **ni trop énergique**, *il maîtrise,* **avec une finesse quasi mozartienne**, *un lyrisme généreux.* (**Neither very romantic, nor very energetic,** he masters, **with great delicatesse as Mozart's**, a generous lyrism.)

**(3)** **Militant mais opportuniste**, **franc-tireur mais habile**, **sociable mais anticonformiste**, *le directeur de l'Opéra de Paris sait manier les paradoxes pour parvenir à ses fins.* (**Militant but opportunist, dynamic but rigourous, sociable but anticonformist**, the Paris Opera's director knows how to handle paradoxs in order to reach his goals.)

Jackiewicz [12] provides about 20 handcrafted linguistic patterns to automatically extract appositive qualifying phrases. Some examples of those patterns[5] are:

- Nominal groups (NG): (det) N de NG
  - *Femme de tête, X* (stubborn woman, X);
  - *X, le maestro de la désinflation* (X, the master of deflation).
- Adverbs: *courageusement, X* (courageously, X);
- Prepositional groups: *en mauvaise posture, X* (in a bad shape, X);
- Adjectival groups: *imprévisible et fantasque, X* (unpredictable and little bit crazy, X);
- Participle groups : *réputé pour son caractère bourru, X* (known for his obstinated personality, X).

Obviously, the definition of those linguistic patterns by hand is a tedious task. It shows the interest of our approach. In the sequel, we describe the process to help the linguistic expert to discover linguistic patterns characterizing qualifying phrases.

### 3.2 Corpora Constitution

As there is no available corpus with qualifying phrases, we have built two corpora.

The first corpus, called AXIOLO, is a set of occurrences obtained with linguistic patterns coming from [12]. Patterns are applied on the articles of the French newspaper *"Le Monde"*, of the topic *"Portrait"* (*i.e.* profile), from July to December of 2002 (884 articles). The building process of this first corpus leads to corpus almost without noise.

The second corpus, called ARTS, is also generated from the French newspaper *"Le Monde"* from articles of the topic *"Arts"* in 2006 (3,539 articles). We first applied the Treetagger tool [18] on the corpus to split sentences in constituents bounded by punctuations[6]. Our method is tolerant regarding Treetagger errors. Actually, if a wrong tag commonly occurs, this tag impacts resulting patterns without disrupting the result quality. Then, we used heuristics to filter out irrelevant constituents from sequential patterns, the ones that have no qualification. For instance, a proposition with a conjugated verb, a circumstantial group of time, of space, a goal, a cause, a condition are irrelevant qualifying phrases. Applying heuristics consists of testing for instance if a verb

[5] In the examples, the $X$ represents the subject of the qualification.

[6] Treetagger is used with the original training set.

occurs in a given phrase, or if there exists a temporal term such as "Monday". The list of irrelevant terms is built according to the Leff lexicons [17]. We also manually added to this list some of typical French expressions such as "*d'une part (from one hand)*", "*en référence (as refered to)*", and so on. Finally, using heuristics allows to remove 113,812 constituents from the 127,388 originals. The resulting corpus is partially noisy. We have manually evaluated 32% of noise from a sample of 1,000 phrases. Table 3 gives the characteristics of corpora.

| Corpus | # seq | # items | # avg. Itemsets per seq. | # max. itemsets in seq. | # avg. Items per seq. | # max. items in seq. |
|---|---|---|---|---|---|---|
| *AXIOLO* | 4,063 | 4,135 | 3 | 17 | 7 | 43 |
| *ARTS* | 13,576 | 20,796 | 6 | 32 | 16 | 89 |

**Fig. 3.** Corpora Characteristics.

### 3.3 Extraction of Sequential Patterns

In order to extract the closed sequential patterns of itemsets, we used our algorithm with both gap (with $g(0,0)$) and *begin_with* constraint (cf. Section 2.2). We have conducted 10 experiments for each corpus with a relative support threshold between 0.05% to 50%. With the AXIOLO corpus, it means that a sequential pattern is frequent as soon as it appears in respectively 2 to 2,031 sequences. With the ARTS corpus, a sequential pattern is frequent as soon as it appears in respectively 6 to 6,788 sequences.

Results indicate that high *minsup* values provide very generic patterns, with only grammatical categories in itemsets (i.e. without lemmas or inflected forms of a term). According to our application on the discovery of linguistic patterns of appositive qualifying phrases, it is more relevant to use a low *minsup* to obtain sequential patterns combining the different levels of word abstraction. However, a low $minsup$ produces an high number of patterns. For instance, with $minsup = 0.05\%$, 8,536 patterns are extracted from the ARTS corpus.

The validation task of patterns is difficult because of the high number of extracted sequential patterns. It shows the interest of our method based on the partial order of patterns and the Camelis tool (cf. Section 2.3). For each sequential pattern, $P$, the set of phrases matched by $P$ and coming from a given corpus are grouped and filtered together. A linguist can then easily check the set of phrases matched by a pattern, it is especially interesting with noisy corpora. Then the validation of sequential patterns as linguistic patterns becomes easier for a linguist.

### 3.4 Experimental Results

**Runtime.** Our first aim is to evaluate the gain of the integration of constraints in the mining algorithm. For that purpose, we measure the runtime of the *Clospan* algorithm proposed in Illimine[7] which is a very competitive prototype, and the runtime of our

[7] http://illimine.cs.uiuc.edu/

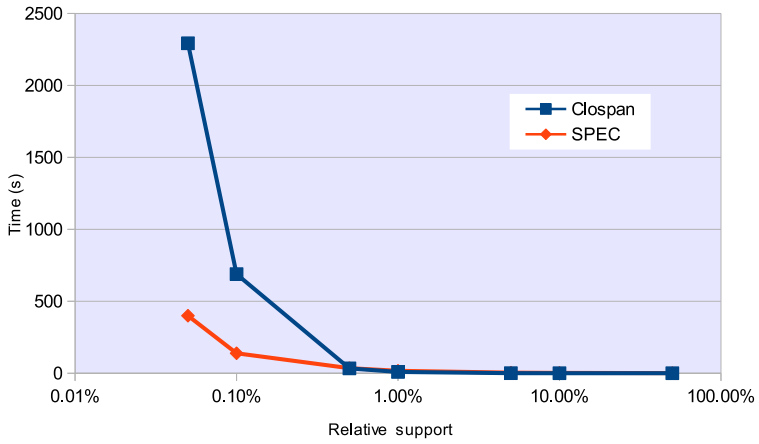


**Fig. 4.** Runtime comparing Clospan and our algorithm: SPEC.

algorithm. We did the process on 10 experiments with the ARTS corpus. Experiments were conducted with an Intel Core 2 Duo Processor T9600 with 8 GHz of RAM.

Figure 4 shows that our algorithm, SPEC (Sequential Pattern Extraction with Constraints) is much faster than *Clospan* for small relative supports. Note that *Clospan* only extracts the frequent closed sequential patterns and it does not integrate the gap constraint neither the *begin_with* constraint. When considering *Clospan*, we should take into account the time needed for the application of the constraints in a post-processing step. Therefore, the whole runtime of the process with *Clospan* would be higher.

**Qualitative results.** Evaluating an unsupervised method is a difficult task. A first way is to compare the results obtained by the method to a reference corpus, but such a corpus can be missing. A second way is to conduct an evaluation with an expert, but it requires a lot of time. In our case, we do not have a reference corpus on appositive qualifying phrases. Thus, we present our experimental results rather on the qualitative way, showing the originality and the usefulness of our method. Its success relies on the joint use of itemsets in sequences to catch several levels of information and the hierarchical property of patterns to validate them.

First, we want to evaluate the interest of sequential patterns made of itemsets. For that purpose, we have conducted on both corpora the mining of sequential patterns only made of items. We consider three kinds of sequences: the lemma, the grammatical category or combining the lemma and the grammatical category of a term. Results indicate that the obtained patterns are very specific or very generic. Examples of specific patterns are: $\langle homme\ de\ conviction \rangle$ $(man\ of\ conviction)$ on lemma sequences ; $\langle homme/NOUN\ de/PRP\ conviction/NOUN \rangle$ on the combinations, which is almost the same sequence as the sequence obtained with lemma. With sequential patterns of items, we have the same level of abstraction for each word. For instance, we can have a generic pattern like $\langle NOUN\ PRP\ NOUN \rangle$ with only

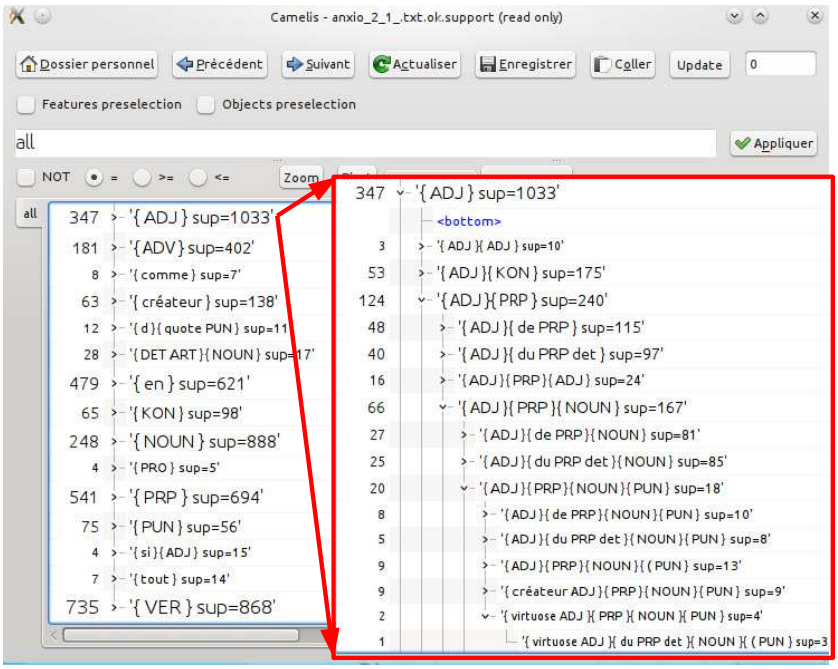


**Fig. 5.** Pattern discovering: Example with the ART corpus.

grammatical categories. Our experiments indicate that patterns with different levels of abstraction can be **only discovered by using itemsets**. For instance, the pattern $\langle(hommes\ homme\ NOUN)\ (de\ PRP)\ (NOUN)\rangle$ uses inflected forms, lemmas, and grammatical categories.

Results on AXOLIO corpus show that our method is able to *automatically* recover all the handcrafted linguistic patterns presented in Section 3.2. Even better, our method **declines generic patterns on different specific ways**. Note that the very specific patterns are obtained by setting a low support threshold. For instance the specific patterns for the generic pattern $\langle(NOUN)\ (PRP)\ (NOUN)\rangle$ are:

$\rightarrow\ \langle(NOUN)\ (du\ PRP\ det)\ (NOUN)\rangle$;
$\rightarrow\ \langle(NOUN)\ (de\ PRP)\ (NOUN)\rangle$;
  $\rightarrow\ \langle(spécialiste\ NOUN)\ (de\ PRP)\ (NOUN)\rangle$;
  $\rightarrow\ \langle(homme\ NOUN)\ (de\ PRP)\ (NOUN)\rangle$;
    $\rightarrow\ \langle(homme\ NOUN)\ (de\ PRP)\ (conviction\ NOUN)\rangle$.

In addition, results produce syntagmatic constructions with various forms and expansions. The example below shows some extracted constructions of adjectival group:

$\langle(ADV)\ (ADJ)\rangle$ ; $\langle(ADV)\ (ADJ)\ (PRP)\ (VER\,infi)\rangle$; $\langle(ADV)\ (ADJ)\ (et\ KON)\ (ADJ)\rangle$ ; $\langle(ADJ)\ (mais\ KON)\ (ADJ)\rangle$ ; $\langle(ADJ)\ (et\ KON)\ (VER\,pper)\rangle$ ; $\langle(ADV)\ (ADJ)\ (\grave{a}\ PRP)\ (VER\,infi)\rangle$ ; $\langle(ADV)\ (ADV)\ (ADJ)\rangle$ ; $\langle(ADV)\ (plus\ ADV)\ (\ ADJ)\rangle$, and so on.

Results on the ARTS corpus show the interest of the method with noisy data. Let us recall that this corpus was automatically generated and the phrases have not been tagged. Therefore, some sequential patterns extracted from the corpus may suggest non relevant patterns. Thanks to the hierarchical navigation proposed in the process by using the Camelis tool, such noisy patterns can be easily removed and the selection of relevant linguistic patterns is easy. Figure 5 depicts an excerpt of the pattern hierarchy within this corpus. Then, we can discover **new linguistic patterns** (compared to those proposed in [12], resulting of a manual extraction) in order to extract qualifying appositive phrases. For instance, we discover the pattern $\langle\ (ADJ)\ (pour)\ (DET)\ (NOUN)\ \rangle$ which matches phrases such as: "*célèbre pour son monastère" ("famous for its monastery")*, "*baroque pour une histoire d'amour" ("baroque for a love story")*). We also discover some variations or extensions: $\langle (ADV)\ (ADJ)\ (pour)\rangle$ (e.g., "*très célèbre pour" ("very famous for")*), $\langle (ADJ)\ (pour)\ (VER)\rangle$ (e.g., "*indispensable pour assurer" ("essential to ensure")*).

## 4 Conclusion

We have proposed an approach based on the extraction of sequential patterns which aims at automatically discovering linguistic patterns. Whereas existing methods are based on single-item sequences, our approach extracts sequences of itemsets. It leads to more expressiveness in the discovered patterns by combining the different levels of word abstraction (word, lemma, grammatical category). In addition, the extracted patterns are understandable by a human unlike machine learning based methods. Moreover, sequence mining approaches are not language-dependent. We have designed an algorithm for mining such sequential patterns. An outstanding idea of our algorithm is to take into account constraints in order to reduce the number of extracted patterns and therefore also to reduce the time processing. However, the number of sequential patterns can remain high. In order to address that problem, we have proposed to take advantage of the partial order between patterns and use a tool allowing a user to easily navigate within the pattern space and validate sequential patterns as relevant linguistic patterns. We have conducted some experiments to discover linguistic patterns to extract appositive qualifying phrases. Results show that even with a noisy corpus. In addition thanks to the navigation tool, an expert can easily select relevant patterns.

Further work is the evaluation of the patterns according to a task without gold standard as the task that we consider in this paper. This is a well-known issue in unsupervised methods. Another further work is to enhance the algorithm with new constraints in order to reduce the number of extracted sequential patterns. For example, a constraint of maximum support can be used to filter out patterns very general. Finally, the approach presented in this paper is not specific to the detection of appositive qualifying phrases. It can also be used to extract other kinds of linguistic patterns, acquiring new resources as lexicons or extraction rules. For instance, mining sequential patterns of itemsets in order to extract the relationships between named entities (such as interaction between genes) would improve the state-of-the-art works.

**Acknowledgements.** This work is partly supported by the ANR (French National Research Agency) funded project Hybride ANR-11-BS02-002.

## References


1. R. AGRAWAL AND R. SRIKANT: *Mining sequential patterns*, in ICDE, IEEE, 1995.
2. F. BONCHI: *On closed constrained frequent pattern mining*, in In Proc. IEEE Int. Conf. on Data Mining ICDM04, Press, 2004, pp. 35–42.
3. M. E. CALIFF AND R. J. MOONEY: *Relational learning of pattern-match rules for information extraction*, in AAAI-99, 1999, pp. 328–334.
4. P. CELLIER, T. CHARNOIS, AND M. PLANTEVIT: *Sequential patterns to discover and characterise biological relations*, in CICLing'10, springer, 2010.
5. B. A. DAVEY AND H. A. PRIESTLEY: *Introduction To Lattices And Order*, Cambridge University Press, 1990.
6. G. DONG AND J. PEI: *Sequence Data Mining*, Springer, 2007.
7. S. FERR: *Camelis: a logical information system to organize and browse a collection of documents*. Int. J. General Systems, 38(4) 2009.
8. W. J. FRAWLEY, G. PIATETSKY-SHAPIRO, AND C. J. MATHEUS: *Knowledge discovery in databases: An overview*, in KDD, AAAI/MIT Press, 1991, pp. 1–30.
9. K. FUNDEL, R. KÜFFNER, AND R. ZIMMER: *RelEx - relation extraction using dependency parse trees*. Bioinformatics, 23(3) 2007, pp. 365–371.
10. C. GIULIANO, A. LAVELLI, AND L. ROMANO: *Exploiting shallow linguistic information for relation extraction from biomedical literature*, in EACL, 2006.
11. J. R. HOBBS AND E. RILOFF: *Information extraction*, in Handbook of Natural Language Processing, Second Edition, N. Indurkhya and F. J. Damerau, eds., CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010.
12. A. JACKIEWICZ: *Structures avec constituants détachés et jugements d'évaluation*. Document Numérique, 13(3) 2010, pp. 11–40.
13. M. KRALLINGER, F. LEITNER, C. RODRIGUEZ-PENAGOS, AND A. VALENCIA: *Overview of the protein-protein interaction annotation extraction task of BioCreative II*. Genome Biology, 9 2008.
14. C. NÉDELLEC: *Machine learning for information extraction in genomics - state of the art and perspectives*, in Text Mining and its Applications: Results of the NEMIS Launch Conf. Series: Studies in Fuzziness and Soft Comp. Sirmakessis, Spiros, 2004.
15. J. PEI, J. HAN, B. MORTAZAVI-ASL, H. PINTO, Q. CHEN, U. DAYAL, AND M. HSU: *Prefixspan: Mining sequential patterns by prefix-projected growth*, in ICDE, IEEE Computer Society, 2001, pp. 215–224.
16. E. RILOFF: *Automatically generating extraction patterns from untagged text*, in AAAI/IAAI'96, 1996.
17. B. SAGOT, L. CLÉMENT, E. DE LA CLERGERIE, AND P. BOULLIER: *The lefff 2 syntactic lexicon for french: architecture, acquisition, use.*, in LREC 06, Głnes, Italie, 2009.
18. H. SCHMID: *Probabilistic part-of-speech tagging using decision trees*, in Proceedings of International Conference on New Methods in Language Processing, September 1994.
19. R. SRIKANT AND R. AGRAWAL: *Mining sequential patterns: Generalizations and performance improvements*, in EDBT, P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, eds., vol. 1057 of LNCS, Springer, 1996, pp. 3–17.
20. J. WANG AND J. HAN: *Bide: Efficient mining of frequent closed sequences*, in ICDE, IEEE Computer Society, 2004, pp. 79–90.
21. X. YAN, J. HAN, AND R. AFSHAR: *Clospan: Mining closed sequential patterns in large databases*, in SDM, D. Barbará and C. Kamath, eds., SIAM, 2003.
22. M. J. ZAKI: *SPADE: An efficient algorithm for mining frequent sequences*. Machine Learning Journal, 42(1/2) Jan/Feb 2001, pp. 31–60, special issue on Unsupervised Learning.