What About Sequential Data Mining Techniques to Identify Linguistic Patterns for Stylistics?

Solen Quiniou, Peggy Cellier, Thierry Charnois, Dominique Legallois

What About Sequential Data Mining Techniques to Identify Linguistic Patterns for Stylistics?

Solen Quiniou[1,2], Peggy Cellier[3], Thierry Charnois[1], and Dominique Legallois[2]

[1]GREYC Université de Caen Basse-Normandie, Campus 2, 14000 Caen
[2]CRISCO Université de Caen Basse-Normandie, Campus 1, 14000 Caen
[3]IRISA-INSA de Rennes, Campus de Beaulieu, 35042 Rennes Cedex

Abstract. In this paper, we study the use of data mining techniques for stylistic analysis, from a linguistic point of view, by considering emerging sequential patterns. First, we show that mining sequential patterns of words with gap constraints gives new relevant linguistic patterns with respect to patterns built on n-grams. Then, we investigate how sequential patterns of itemsets can provide more generic linguistic patterns. We validate our approach from a linguistic point of view by conducting experiments on three corpora of various types of French texts (*Poetry*, *Letters*, and *Fiction*). By considering more particularly poetic texts, we show that characteristic linguistic patterns can be identified using data mining techniques. We also discuss how to improve our proposed approach so that it can be used more efficiently for linguistic analyses.

1 Introduction

The study of phraseology - including stylistics - is a research field that has been investigated over the past 30 years by the linguistic community. More recently, there has been a particular interest in studies from corpus linguistics. Two main approaches can be identified: corpus-based and corpus-driven. *Corpus-based* approaches assume the existence of linguistic theories and use corpora to analyze their application and hence to validate them. *Corpus-driven* approaches consider that linguistic constructs emerge from corpus analysis. This analysis allows the discovery of co-occurring word patterns that will be the basis of linguistic analyses. Our work is part of the corpus-driven approaches since our goal is to assist linguists in discovering new linguistic constructs without any prior knowledge.

One of the first corpus-driven approach was proposed by Renouf and Sinclair [1]. It consists on a study of collocational frameworks thanks to corpora; *collocational frameworks* represent discontinuous sequences of two grammatical words enclosing a lexical word (*e.g.*, *“ many + ? + of “* that means *many* followed by a variable lexeme - symbolized by *?* - itself followed by *of*). However, this approach is not entirely corpus-driven since the studied collocational frameworks were pre-selected by Renouf and Sinclair. In fact, most of the so-called corpus-driven approaches are partly corpus-based [1]. More recently, Biber presented an

[1] See [2] for a more detailed state of the art on corpus-driven approaches.

interesting approach, entirely corpus-driven, to identify frequent patterns from corpora [2]. To do so, he relies both on a preliminary work on the identification of *lexical bundles* (*i.e.*, frequent sequences of contiguous words, aka n-grams) and on collocational frameworks to identify fixed and variable elements in the patterns he extracted. Furthermore, Biber considers two language registers (conversation and academic writing) and shows the interest of using a corpus-driven approach to study the specificities of patterns appearing in each register.

In this paper, we present a first and original study which aims at showing the interest of data mining methods for the stylistic analysis of large texts. The goal is to provide to the linguist experts some prominent, relevant, and understandable patterns which can be characteristic of a specific type of text so that these experts can carry out a stylistic analysis based on these patterns. In fact, our work is in the continuity of Biber's but we consider various text types (instead of language registers) that we study from a stylistic point of view. To do so, we set up a methodology based on sequential data mining, from the extraction of patterns to the selection of the most relevant. We apply this methodology to stylistics. To the best of our knowledge, data mining methods have not yet been used in the field of stylistics whereas one of their advantages is to offer an interpretable result to users, as opposed to numerical methods such as Hidden Markov Models or Conditional Random Fields. Indeed, the latter methods have been shown to achieve good results for tasks like text categorization or information extraction but they produce outputs hardly understandable by humans. Thus, the approach that we propose is based on *frequent sequential patterns* [3], a well-known data mining technique to automatically discover frequent patterns based on the sequential order of data. We consider two types of sequential patterns: single-item patterns (an *item* represents a single piece of information, *e.g.* a word form); and *itemset* patterns. In this second type of patterns, a word is represented by a set of features. Therefore, extracted itemset patterns may combine different levels of abstraction (word forms, lemmas, POS tags, etc.): for instance, $\langle (PREP)\ (DET)\ (NC) \rangle$ or $\langle (to)\ (the\ DET)\ (NC) \rangle$[2]. Furthermore, as we set our study in the field of stylistics, the end-goal is to extract patterns that are characteristic of a certain type of text. This is the reason why we focus on a specific type of sequential patterns: *emerging patterns*. Emerging patterns can capture contrast characteristics between classes or datasets [4]. Furthermore, these patterns can be analyzed by experts to discover new relationships in a given domain for a better understanding of it. Here, extracted emerging patterns could then be analyzed by linguists to discover linguistic patterns, characteristic of a certain type of text.

The rest of this paper is organized as follows. First, our methodology based on sequential data mining is introduced in Section 2. Then, Section 3 presents experimental results on the use of our methodology for stylistics both from a quantitative and a linguistic point of view. Finally, Section 4 discusses the leads to further investigate, while Section 5 draws some conclusions.

[2] PREP, DET, and NC are the POS tags for prepositions, determiners, and common nouns respectively.

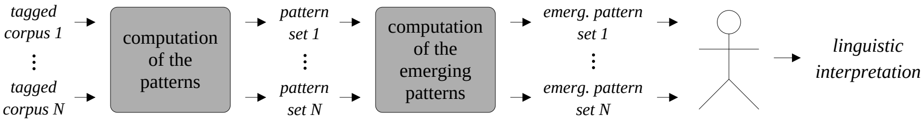

Fig. 1. Overview of our proposed approach

2 Methodology

In this section, we give an overview of the proposed approach to identify characteristic linguistic patterns for each type of text (Section 2.1). Then, we present the sequential data mining techniques on which our approach is based: frequent sequential patterns (Section 2.2) and emerging patterns (Section 2.3).

2.1 Overview of the Proposed Approach

Figure 1 illustrates the various steps of our approach. N corpora are used as the inputs of the process, with one corpus corresponding to a considered type of text. Each corpus is first pre-processed and then all its words are labeled with their lemma and their POS category (see Section 3.1). In the first step of our approach, sequential patterns are extracted for each corpus: N sets of sequential patterns are therefore obtained. Then, in the second step, sets of emerging patterns are selected for each corpus, from the N sets of sequential patterns previously extracted. Lastly, the N sets of emerging patterns are given to a linguist so that he can use them to perform a linguistic interpretation. The first and second steps are presented in greater details in the next sub-sections.

2.2 Sequential Pattern Mining

Sequential pattern mining is a well-known data mining technique used to find regularities in sequence databases, by considering the temporal order of the data. This technique was introduced by Agrawal *et al.* in [3].

An *itemset*, I, is defined as a set of literals called *items*, denoted by $I = (i_1 \dots i_n)$. For example, $(a\ b)$ is an itemset with two items: a and b. A *sequence*, S, is defined as an ordered list of itemsets, denoted by $S = \langle I_1 \dots I_m \rangle$. For instance, $\langle (a\ b)(c)(d)(a) \rangle$ is a sequence of four itemsets. It should be noted that a lot of applications need only one item in their itemsets (*e.g.* DNA strings or protein sequences). These particular kinds of sequences are called *single-item sequences*; for the sake of clarity, they are denoted by $S = \langle i_1 \dots i_n \rangle$, where $i_1 \dots i_n$ are items. Several algorithms have been developed to efficiently mine that kind of specific sequences, for example [5]. In the rest of the paper, both kinds of sequences will be considered, *i.e.* single-item sequences and itemset sequences. 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 thus called a *subsequence* of S_2, which is noted $S_1 \preceq S_2$. For

Table 1. SDB_1: a sequence database

Sequence identifier	Sequence
1	$\langle(a\ b)(c)(d)(a)\rangle$
2	$\langle(d)(a)(e)\rangle$
3	$\langle(d)(a\ b\ e)(c\ d\ e)\rangle$
4	$\langle(c)(a)\rangle$

example, we have the following relation: $\langle(c)(a)\rangle \preceq \langle(a\ b)(c)(d)(a)\rangle$. 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 represents 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 containing S_1 in the database. For example, in Table 1, $sup(\langle(a)(e)\rangle) = 2$ since sequences 2 and 3 contain an itemset with a followed by an itemset with e. The *relative support* of sequences may also be used, as defined by Equation 1:

$$sup(S_1) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \wedge (S_1 \preceq S)\}|}{|SDB|} \qquad (1)$$

A *frequent pattern* is a sequence such that its support is greater or equal to a given threshold: $minsup$. Sequential pattern mining algorithms thus extract all the frequent sequential patterns that appear in a sequence database.

Because the set of frequent sequential patterns can be very large, there exists a condensed representation which eliminates redundancies without loss of information: *closed sequential patterns* [6]. A frequent sequential pattern S is closed if there exists no other frequent sequential pattern S' such that $S \preceq S'$ and $sup(S) = sup(S')$. For instance, with $minsup = 2$, the sequential pattern $\langle(b)(c)\rangle$ from Table 1 is not closed whereas the pattern $\langle(a\ b)(c)\rangle$ is closed. Moreover, in order to drive the mining process towards the user objectives and to eliminate irrelevant patterns, one can define constraints [7,8]. The most commonly used constraint is the frequency constraint (that assigns a value to $minsup$). Another widespread constraint is the gap constraint. A sequential pattern with a gap constraint $[M, N]$, denoted by $P_{[M,N]}$, is a pattern such as at least $M - 1$ itemsets and at most $N - 1$ itemsets are allowed between every two neighbor itemsets, in the original sequences. For instance, let $P_{[1,3]} = \langle(c)(a)\rangle$ and $P_{[2,3]} = \langle(c)(a)\rangle$ be two patterns with two different gap constraints and let us consider the sequences of Table 1. Sequences 1 and 4 are occurrences of pattern $P_{[1,3]}$ (sequence 1 contains one itemset between (c) and (a) whereas sequence 4 contains no itemset between (c) and (a)), but only sequence 1 is an occurrence of $P_{[2,3]}$ (only sequences with one or two itemsets between (c) and (a) are occurrences of this pattern).

In this paper, the considered databases correspond to corpora. Furthermore, two kinds of sequential patterns are considered: single-item patterns and itemset patterns. In that last case, itemsets can be made up of three types of items: word forms, lemmas, and POS tags.

2.3 Emerging Patterns

Emerging patterns are defined as sequential patterns whose support increases significantly from one dataset to another one [4]. More specifically, emergent patterns are sequential patterns whose *growth rate* - the ratio of the supports in the two datasets - is larger than a given threshold: ρ. Thus, a sequential pattern P from a dataset D_1 is an *emerging pattern* to another dataset D_2 if $GrowthRate(P) \geq \rho$, with $\rho > 1$ and with $GrowthRate(P)$ being defined by:

$$GrowthRate(P) = \begin{cases} \infty, & \text{if } sup_{D_2}(P) = 0 \\ \frac{sup_{D_1}(P)}{sup_{D_2}(P)}, & \text{otherwise} \end{cases} \tag{2}$$

with $sup_{D_1}(P)$ ($sup_{D_2}(P)$ respectively) being the relative support of the pattern P in D_1 (D_2 respectively). Since we are only interested in patterns belonging to D_1, we do not consider patterns P with $sup_{D_1}(P) = 0$.

In the case of stylistic analyses, each dataset contains the frequent sequential patterns of a corpus and thus of the corresponding type of text. It corresponds to the patterns extracted during the first step of our approach (see Section 2.2). Because we consider more than two types of text, we compute the emerging patterns of a considered type of text with respect to every other type, according to Equation 2. Finally, only the patterns that are emerging to every other type of text are kept as emerging patterns for a considered type of text. The computation of all the emerging patterns is done efficiently based on [9].

3 Experimental Evaluation

In this section, we report the results of our experimental evaluation on using sequential pattern mining techniques for stylistics. First, in Section 3.1, we describe the used corpora as well as the setup of the various parameters used to extract emerging sequential patterns. Then, we present an analysis of the extracted sequential patterns, at two levels: from a quantitative point of view (in Section 3.2), and from a linguistic point of view for stylistics (in Section 3.3).

3.1 Experimental Setup

Corpora We created three corpora, corresponding to various types of text: *Poetry*, *Letters*, and *Fiction*. To build each corpus, we selected all the texts of the 1800-1900 era - provided by the French resources of the CNRTL[3] - corresponding to the considered type of text. For example, authors from *Poetry* include Lamartine and Musset, whereas Hugo and Lamennais are part of the authors of *Letters*, and Chateaubriand and Zola are authors of *Fiction*. Then, these three corpora were pre-processed. The pre-processing steps consisted in setting the words in lower-case, and then splitting the texts into sequences at punctuation marks of the set: {*’.’, ’?’, ’!’, ’. . . ’, ’;’, ’:’, ’,’*}. Table 2 gives some details on each corpus: the number of authors, of works, of sequences, and of words.

[3] Centre National des Ressources Textuelles et Linguistiques : www.cntrl.fr

Table 2. Characteristics of the corpora *Poetry*, *Letters*, and *Fiction*

Corpus	#authors	#works	#sequences	#words
Poetry	27	48	151 116	1 167 422
Letters	5	9	234 997	1 562 543
Fiction	37	52	663 860	5 105 240

After being pre-processed, the corpora were POS tagged using *Cordial*[4], a tagger that is known to outperform TreeTagger on French texts. Thus, each word of the corpora was associated with its form, its lemma and its POS tag. After first experimentations, it turns out that the POS tags given by Cordial were too much specific; we thus post-processed them to reduce their number (as a consequence, it reduces the number of extracted patterns). First, too specific categories were merged into more general ones. For example, the adjective category was initially decomposed into 16 categories (depending on the gender, the number, or whether the word starts with a mute *h* letter). Thus, the following categories were created, to replace their corresponding sub-categories: adjectives (ADJ), determiners (DET), common nouns (NC), proper nouns (NP), demonstrative pronouns (PD), relative pronouns (PR), indefinite pronouns (PI), and past participles (VPARP). Then, categories corresponding to personal pronouns were decomposed into 2 tags: one for the personal pronoun (PPER), and one for the person (*e.g.* 1S for the singular first person). Moreover, categories corresponding to verbs were decomposed into 3 tags: one for the verb (V), one for the mode of the verb (*e.g.* INDP for the present of the indicative mode), and one for the person (the same ones as for the personal pronouns). At the end, we had a set of 35 tags instead of the 133 initial tags. Using this new set of tags, the phrase *"a rose that we smell"* is translated as $<$ *(a a DET) (rose rose NC) (that that PR) (we we PPER 1P) (smell smell V PRES 1P)* $>$.

Mining Single-Item Sequences First, we considered single-item sequences of words. To perform the mining task on the three corpora, we used dmt4 [5] that allows the definition of various constraints on the extracted single-item sequential patterns: the length, the frequency (by setting *minsup*, the support threshold), or the gaps (by choosing the values of $[M, N]$). We set the length of the patterns to be between 2 and 20. We chose the value of *minsup* empirically as a trade-off between having interesting patterns with a low support (thus setting a low value to *minsup*) and having not too many patterns (thus setting a high value to *minsup*). Because of the differences in the corpora sizes (*Fiction* is five times bigger than *Poetry*), we chose a relative threshold whose value is 0.001 %; it corresponds to the following absolute thresholds: 16 for *Poetry*, 12 for *Letters*, and 51 for *Fiction*. That means that only patterns appearing in at least 16 sequences are kept for *Poetry*, for example. For the gap constraints, we chose to consider different values in the following experiments (see Section 3.2): $[1, 1]$,

[4] The Cordial tagger is developed by Synapse Développement (www.synapse-fr.com).

Table 3. Number of patterns and ratio of emerging ones (in brackets) for the corpora

Corpus	Single-item patterns with gaps				Itemset
	$[1,1]$	$[1,2]$	$[1,3]$	$[1,5]$	patterns
Poetry	18 816	37 933	55 762	86 901	2 245 326
	(30.7 %)	(27.0 %)	(24.3 %)	(22.6 %)	(11.4 %)
Letters	16 936	36 849	56 755	96 549	10 128 288
	(50.2 %)	(50.7 %)	(50.4 %)	(50.0 %)	(57.4 %)
Fiction	78 210	175 645	282 967	512 647	11 681 913
	(6.1 %)	(5.3 %)	(4.9 %)	(4.6 %)	(71.2 %)
Total	113 962	250 427	395 484	696 097	24 055 527
	(16.7 %)	(15.3 %)	(14.2 %)	(13.2 %)	(59.8 %)

$[1,2]$, $[1,3]$, and $[1,5]$. It is worth noting that the $[1,1]$ gap constraint corresponds to considering n-gram patterns. Indeed, patterns extracted under this constraint correspond to sub-sequences of consecutive words of the corpus.

Mining Itemset Sequences Finally, we considered itemset sequences, where each itemset represents a word with its form, its lemma, and its POS tag. To mine these itemset sequences, we chose CloSpan [6] that extracts closed sequential itemset patterns. CloSpan allows to set only one constraint: the support threshold *minsup*. We also chose empirically the value of *minsup* to be 0.15%. Note that, because no gap constraint can be set in CloSpan, we had to choose a higher value for *minsup* to limit the total number of patterns that are generated and hence to limit the computation time. The drawback of that choice is that interesting patterns may not be extracted because their support may be too low (for example, the absolute support threshold is 1 000 for *Fiction*).

Selecting Emerging Patterns To select the emerging patterns of the corpora, we set the threshold ρ just above 1: $\rho = 1.001$. This threshold is used on both single-item patterns and itemset patterns.

3.2 Quantitative Analysis of the Patterns

In this sub-section, we present quantitative results on the single-item patterns and on the itemset patterns. The set of extracted patterns being large, this quantitative analysis allows us to select the patterns that will be actually analyzed from a linguistic point of view, for the stylistic task (see Section 3.3).

Table 3 gives the number of extracted patterns for the three corpora, by considering the two types of patterns: single-item patterns (with various gap constraints) and itemset patterns. The ratio of emerging patterns is also given for each type of patterns. Thus, among the 18 816 patterns extracted from *Poetry* (by setting the gap constraint to $[1,1]$), 30.7 % of the patterns are emerging ones (corresponding to 5 776 patterns). First, we can see that selecting emerging

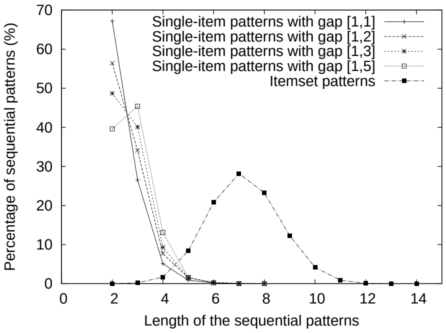

Fig. 2. Distribution of the emerging patterns w.r.t. the length

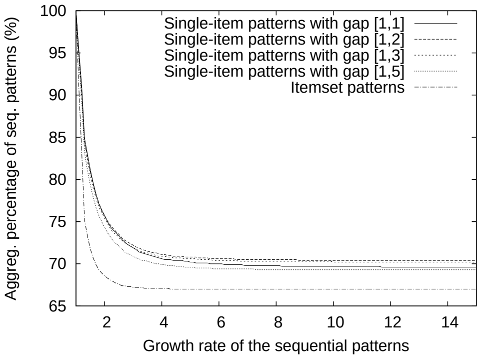

Fig. 3. Distribution of the emerging patterns w.r.t. the growth rate

patterns allows a large reduction of the total number of sequential patterns to analyze. Moreover, it allows to focus our attention on more interesting patterns in the context of stylistics. That is why we will only consider emerging patterns in the rest of the analyses. Furthermore, we can see that by increasing the gap constraint, the rate of single-item emerging patterns tends to decrease: it means that additional extracted patterns tend to be non-specific patterns of the studied types of text. For the stylistic analysis presented in Section 3.3, we set the gap constraint to $[1,3]$ as a tradeoff between the total number of extracted single-item patterns and their relevance. Finally, we can see that many more itemset patterns are extracted, compared to the number of extracted single-item patterns.

Then, we study the distribution of the emerging patterns w.r.t. their length. Figure 2 plots the relative number of patterns for the various pattern lengths, for the single-item patterns (the length is given as the number of items) and for the itemset patterns (the length is given as the number of itemsets). The pattern distributions are computed on the three corpora considered as a whole, for each gap constraint value considered. We can see that most of the single-item patterns contain between 2 and 5 items whereas most of the itemset patterns contain between 4 and 11 itemsets. Therefore, itemset patterns represent longer linguistic patterns. Moreover, there are a lot of single-item patterns of length 2 but they are not as instructive as longer patterns - from a linguistic point of view. That is why we will only consider patterns whose length is greater than 2, for the stylistic analysis.

Finally, we study the distribution of the emerging patterns w.r.t. growth rates. Figure 3 plots the aggregate relative number of emerging patterns as a function of the growth rate, by considering the three corpora as a whole. It means that, for example, 67.1 % of the emerging itemset patterns have a growth rate greater than 4. We can see that most of the emerging patterns have an infinite growth rate as the aggregate rate of emerging patterns is stable for growth rates greater than 10. It means that most of the emerging patterns appear only in a certain type of text (and not at all in the other types of text). In the stylistic analysis, we consider only emerging patterns with an infinite growth rate.

Finally, only itemsets containing both POS tags and word forms or lemmas are considered during the stylistic analysis. Patterns containing only POS tags are therefore removed as they are too general and patterns containing only word forms or lemmas are also removed as they are too specific. In fact, most of the itemset patterns contain both POS tags and words since these patterns represent 93.5 % of all the itemset patterns. That concurs with Biber's conclusions [2] on the extracted patterns that contain both variable and fixed elements (patterns only with POS tags thus contain only variable elements whereas patterns only with words contain only fixed elements).

3.3 Stylistic Analysis of the Emerging Patterns

In this sub-section, we present a stylistic analysis of some extracted emerging patterns. We focus our attention more particularly on the *Poetry* corpus.

First of all, we consider single-item patterns. By studying them, we can find some interesting patterns, characteristic of *Poetry*. Table 4 shows examples of such identified characteristic patterns. In the patterns, the symbol * is used to represent a gap of one or more words[5]. Furthermore, we also illustrate each pattern with examples of underlying sequences in *Poetry*. The extracted patterns allow the observation of schematic grammatical structures that are relatively lexicon-independent. Indeed, fixed elements of these patterns are grammatical words whereas variable elements (*i.e.*, filling the gaps) are generally lexical words (*e.g.*, nouns, verbs, or adjectives). We also show the interest of gap constraints that are given as intervals. The pattern *"some*more*than"* allows the identification of two sequences, among others, where the first gap is filled with a different number of words (see Table 4): in the first one, the word *bites* fills the gap whereas it is filled with the words *angular rocks* in the second sequence. This illustrates the generalization capability of single-item patterns with gap constraints (w.r.t. n-gram patterns, for instance).

[5] Symbol * corresponds to symbol *?* used in [1]. Note that symbol * is also used in [2] but it represents a single variable lexeme whereas, in our approach, this symbol represents a gap of one or more words.

Table 4. Examples of characteristic single-item patterns from *Poetry*

Single-item pattern	Example (*with English translation*)
des*plus*que	il a **des** morsures **plus** venimeuses **que** celles de ta bouche
(some*more*than)	(*he has* ***some*** *bites* ***more*** *venomous* ***than*** *those from your mouth*)
	des cailloux anguleux **plus** brillants **que** des marbres
	(***some*** *angular rocks brighter* ***than*** *some marbles*)
on*et*on	une rose qu'**on** respire **et** qu'**on** jette
(we*and*we)	(*a rose that* ***we*** *smell* ***and*** *that* ***we*** *throw*)
	sur des tombeaux divins qu'**on** brise **et** qu'**on** insulte ?
	(*on divine tombs that* ***we*** *break* ***and*** *that* ***we*** *insult?*)
le/la/l'*qui*et*qui	**la** nuit **qui** m'oppresse **et qui** trouble mes yeux
(the*that*and*that)	(***the*** *night* ***that*** *oppresses me* ***and that*** *troubles my eyes*)
	le grelot **qui** résonne **et** le troupeau **qui** bêle
	(***the*** *bell* ***that*** *resounds* ***and*** *the flock* ***that*** *bleats*)
le*du*qui*dans	**le** vent **du** soir **qui** meurt **dans** le feuillage
(the*of the*that*in)	(***the*** *wind* ***of the*** *night* ***that*** *dies* ***in*** *the foliage*)
	le bruit **du** vieux **qui** bêche **dans** la nuit
	(***the*** *sound* ***of the*** *old* ***that*** *digs* ***in*** *the night*)
est*un*qui	**est**-ce **un** goéland **qui** bat de l'aile ?
(is*a*that)	(***is*** *it* ***a*** *gull* ***that*** *flaps its wing?*)
	ta grâce **est** comme **un** luth **qui** vibre au fond du bois
	(your grace ***is*** *like* ***a*** *lute* ***that*** *vibrates deep in the wood*)

Table 5 gives the correspondence between the single-item patterns presented in Table 4 and their associated itemset patterns. First, it can be seen that several itemset patterns may correspond to the same single-item pattern. Furthermore, extracted itemset patterns allow to obtain the POS categories of the variable elements. Therefore, in the context of a stylistic study of types of text, the work of linguists consists in selecting relevant patterns among automatically extracted itemset patterns: this directly gives them grammatical patterns characteristic of a considered type of text.

In fact, the grammatical patterns we consider correspond to *collocational frameworks* in the sense of Renouf and Sinclair [1], *i.e.* collocations on grammatical units and not on lexical units. However, as opposed to their work, we do not chose *a priori* the patterns that are then studied but we automatically discover them from corpora. We can also compare our work to Biber's [2] - who works also on collocational frameworks - but there are some differences. Indeed, our approach allows to directly extract single-item patterns with gaps as well as itemset patterns (corresponding to grammatical patterns) whereas Biber first extracts frequent sequences from corpora and then analyze them one by one to identify variable and fixed elements to finally build various types of patterns that he studies afterwards. Since Renouf and Sinclair paper, works on collocational frameworks have been done in English corpus linguistics, but not in French. Nonetheless, the analysis of collocational frameworks can be full of

Table 5. Grammatical patterns corresponding to some identified single-item patterns

Single-item pattern	(English translation)	Grammatical pattern
des*plus*que	(some*more*than)	some N more ADJ than
on*et*on	(we*and*we)	N that we V and that we V
le/la/l'*qui*et*qui	(the*that*and)	the N that V and (that) V the N that V and the N that V
le*du*qui*dans	(the*of the*that*in)	the N of the N that V in the N
est*un*qui	(is*a*that)	is it a N that V is like a N that V

insights when associated to an actual usage theory considering that grammatical forms come from a linguistic usage (*i.e.* corpus-driven approaches) and are not the result of integrated rules (*i.e.* corpus-based approaches). Therefore, it is interesting to have approaches that automatically extract patterns to provide these collocation frameworks, as it is the case with our proposed approach.

4 Discussion

In the previous section, we have shown that sequential patterns can be interpreted by linguists for stylistic analyses. However, a huge number of sequential patterns are extracted with data mining techniques, from which the interesting ones have to be identified. In this section, we discuss the improvements that could be brought to our current approach to make it easier for linguists to deal with the presented sequential patterns. To this end, we identified two leads.

First, in order to focus our attention on the interesting sequential patterns, it is necessary to be able to set new constraints during the data mining process to narrow the number of extracted patterns down. Thus, it would be interesting to also set gap constraints on itemset patterns (as it is already the case for single-item patterns). In addition, as we can set a minimum threshold, *minsup*, for the pattern supports, it would be interesting to set a maximum threshold, *maxsup*, for the pattern supports as well. Indeed, most interesting sequential patterns generally appear in few sequences. Thus, by not considering too frequent sequential patterns, the total number of patterns would be reduced (for instance, by setting $maxsup = 50$, 21.2 % of the *Poetry* single-item patterns would not be extracted). Moreover, it would allow us to set *minsup* to a lower value, and hence to discover rarer sequential patterns without increasing the total number of patterns. In addition, membership constraints on a certain item type could also be defined to filter out more sequential patterns (*e.g.* only considering sequential patterns containing at least one verb).

Lastly, it would be of interest to provide tools allowing the ordering of the patterns, their filtering, or their exploration jointly with the sequences of the corpus they refer to. Therefore, it would be easier for linguists, in particular, to analyze the extracted sequential patterns (more particularly for itemset patterns or for sequential patterns with gap constraints).

5 Conclusion

In this paper, we have presented a first study on using data mining techniques for stylistics by proposing a methodology based on the extraction of sequential patterns and applied to stylistics. Thus, we have considered two types of sequential patterns: single-item patterns and itemset patterns (based on word forms, lemmas and POS tags). Moreover, we focused our attention on specific sequential patterns: emerging patterns. A quantitative analysis of the sequential patterns extracted from three corpora (representing various types of text, aka *Poetry*, *Letters*, and *Fiction*) has shown that sequential patterns are more powerful than n-grams to express linguistic patterns. That has been confirmed by a linguistic analysis of the extracted emerging sequential patterns since some grammatical patterns characteristic of *Poetry* were identified from these sequential patterns. We also compared our methodology to the one proposed by Biber [2] by showing that ours allows to directly obtain patterns characteristic of types of text. Lastly, we have discussed the improvements that could be brought to our proposed approach both by limiting the total number of extracted sequential patterns and hence to analyze (by defining new constraints on the patterns), and by making it easier for linguists to explore and analyze the patterns (by developing suitable tools for this task). Therefore, these discussions give us leads to investigate in future studies; some of these works are already in progress.

Acknowledgments

This work is partly supported by the French Région Basse-Normandie and by the ANR (National Research Agency) funded project Hybride ANR-11-BS02-002.

References

1. Renouf, A., Sinclair, J.: Collocational Frameworks in English. In: English Corpus Linguistics: Studies in Honour of Jan Svartvik. Longman (1991) 128–143
2. Biber, D.: A corpus-driven approach to formulaic language in English. International Journal of Corpus Linguistics **14** (2009) 275–311
3. Agrawal, R., Srikant, R.: Mining sequential patterns. (In: Proc. of ICDE'95) 3–14
4. Dong, G., Li, J.: Efficient minig of emerging patterns: Discovering trends and differences. (In: Proc. of SIGKDD'99) 43–52
5. Nanni, M., Rigotti, C.: Extracting trees of quantitative serial episodes. (In: Proc. of KDID'07) 170–188
6. Yan, X., Han, J., Afshar, R.: Clospan: Mining closed sequential patterns in large databases. (In: Proc. of SDM'03)
7. Dong, G., Pei, J.: Sequence Data Mining. Springer (2007)
8. Ng, R., Lakshmanan, L., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. (In: Proc. of SIGMOD'98) 13–24
9. Plantevit, M., Crémilleux, B.: Condensed representation of sequential patterns according to frequency-based measures. (In: Proc. of IDA'09) 155–166