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

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 state-of-the-art n-grams. Then, we investigate how sequential patterns of itemsets can provide more generic linguistic patterns. We validate this 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 our current approach based on sequential pattern mining can be improved to be used more efficiently for linguistic analyses.

1 Introduction

The stylistic study of text collections is a research field that has been investigated over the past 30 years by the linguistic community. Nevertheless, computational techniques have been hardly used or remain simple. Usually, they consist in searching for collocations (i.e., co-occurrence relationships between words) that are generally limited to two words [1, 2]. More recently, Biber has presented an interesting study [3], using lexical bundles (corresponding to frequent sequences of contiguous words, aka n-grams) and extending them by allowing variable lexical slots in the sequences (for instance, “to the ? of”, where the symbol ? stands for a variable lexical slot). Thanks to the extended lexical bundles, he identified multi-word characteristic patterns for conversation and for academic writing (each identified pattern is characteristic of either conversation, or academic writing).

In this paper, we present a first and original study which aims at showing the interest of data mining methods for 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. To do so, we set up and experiment a methodology based on sequential data mining, from the extraction of relevant patterns to their selection. To the best of our knowledge, data mining methods have not yet been used in the field of stylistics. Data mining [4] allows implicit, previously unknown, and potentially useful information to be extracted from data [5]. Thus, using data mining techniques is more suitable to the extraction of patterns that can be interpreted by linguists than using numerical text
mining methods (e.g. Hidden Markov Models, Support Vector Machines, Conditional Random Fields...). Indeed, although such numerical techniques have been shown to give good results for tasks like text categorization or information extraction [6], they produce outputs that are not really understandable by users, which is our main requisite. Thus, the approach that we propose is based on frequent sequential patterns [4], a well-known data mining technique to automatically discover frequent patterns. Sequential patterns constitute a more powerful paradigm than n-grams since n-grams can be seen as specific instances of sequential patterns. Indeed, items (i.e. words) within sequential patterns may not necessarily be contiguous. Another interest of sequential patterns lies in the use of sequences of itemsets instead of sequences of items. It means that a word can be represented not only by a single piece of information (e.g. its lemma or POS tag), but by a set of features. Therefore, sequential patterns extracted may combine different levels of abstraction (word forms, lemmas, POS tags...) which provides more or less generic patterns: for instance, \(((\text{DET}) \ (N))\) or \(((\text{to}) \ (the\ \text{DET}) \ (N))\). 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 more specific type of sequential patterns: emerging patterns. Emerging patterns have been widely used in knowledge discovery from databases as they can capture contrast characteristics between classes or datasets [7]. 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, sequential pattern mining and emerging patterns are introduced in Section 2 as well as our methodology based on them. Then, Section 3 presents experimental results on the use of sequential pattern mining for stylistics both from a quantitative and a linguistic point of view. Finally, Section 4 discusses the leads to further investigate to pursue this first study, while Section 5 draws some conclusions.

2 Methodology

In this section, we present our approach based on sequential data mining to provide emerging sequential patterns to linguists, from which they can identify characteristic linguistic patterns for each type of text. First, an overview of the approach is given in Section 2.1 and then, sequential data mining and emerging patterns are presented in Sections 2.2 and 2.3, respectively.

2.1 Overview of the Proposed Approach

Figure 1 illustrates the various steps of our approach. The inputs are corpora that are pre-processed to associate words with their lemma and POS category (see Section 3.1). Here, each corpus corresponds to a certain type of text. In the first step of our approach, sequential patterns are discovered for each corpus, using
data mining techniques. \(N\) sets of sequential patterns are therefore obtained. Then, in the second step, sets of emerging patterns are selected for each corpus, from the sequential patterns previously extracted from the corpora. Lastly, the \(N\) sets of emerging patterns can be interpreted by linguists, what corresponds to the final step of our approach. The first and second steps are presented in greater details in the next sub-sections whereas the tuning of their parameters as well as the pre-processing of the corpora are described in Section 3.1.

### 2.2 Sequential Pattern Mining

*Sequential pattern mining* is a well-known data mining technique introduced by Agrawal et al. in [4] to find regularities in sequence databases. There exist a lot of algorithms to extract sequential patterns [8,9,10,11].

An *itemset*, \(I\), is defined as a set of literals called *items*, and is denoted by \(I = (i_1 \ldots i_n)\). 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 \(S = \langle I_1 \ldots I_m \rangle\). For instance, \(\langle (a\ b) (c) (d) (a) \rangle\) is a sequence of four itemsets. Note that a lot of applications need only one item in their itemsets (e.g. DNA strings, protein sequences). That particular kind of sequence is called a *single-item sequence*; for the sake of clarity, they are denoted by \(S = \langle i_1 \ldots i_a \rangle\), where \(i_1 \ldots i_a\) are items.

Several algorithms have been developed to efficiently mine that kind of specific sequences ([12,10]). In the rest of the paper, both kinds of sequences will be considered, i.e. single-item sequences and itemset sequences.

A sequence \(S_1 = \langle I_1 \ldots I_n \rangle\) is *included* in a sequence \(S_2 = \langle I'_1 \ldots I'_m \rangle\) if there exist integers \(1 \leq j_1 < \ldots < j_n \leq m\) such that \(I_1 \subseteq I'_{j_1}, \ldots, I_n \subseteq I'_{j_n}\). The sequence \(S_1\) is thus called a *subsequence* of \(S_2\), which is noted \(S_1 \preceq S_2\). For example, we have the following relation: \((c(a)) \preceq (a\ b) (c) (d) (a))\). 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, \(sup((c(a))) = 2\) since sequences 1 and 4 contain \((c(a))\). The *relative support* may also be used, as defined by Equation 1:

\[
sup(S_1) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \land (S_1 \preceq S)\}|}{|SDB|} \tag{1}
\]

A *frequent sequential pattern* is a sequence such that its support is greater or equal to the support threshold \(\text{minsup}\). Sequential pattern mining algorithms
Table 1. SDB1: a sequence database

<table>
<thead>
<tr>
<th>Sequence ID</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>⟨⟨a b)(c)(d)(a)⟩⟩</td>
</tr>
<tr>
<td>2</td>
<td>⟨⟨d)(a)(e)⟩⟩</td>
</tr>
<tr>
<td>3</td>
<td>⟨⟨d)(a b e)(c d e)⟩⟩</td>
</tr>
<tr>
<td>4</td>
<td>⟨⟨c)(a)⟩⟩</td>
</tr>
</tbody>
</table>

extract all the regularities that appear in a sequence database by computing its frequent sequential patterns.

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 [9]. A frequent sequential pattern S is closed if there exists no other frequent sequential pattern S' such that S ≤ S' and sup(S) = sup(S'). For instance, with minsup = 2, the sequential pattern ⟨⟨b)(c)⟩⟩ from Table 1 is not closed whereas ⟨⟨a b)(c)⟩⟩ is closed. Moreover, in order to drive the mining process towards the user objectives and to eliminate irrelevant patterns, one can define constraints [13,14]. 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}, means that between every two itemsets of P_{M,N} there exists a gap g(M, N) of at least M - 1 words and at most N - 1 words. For instance, let P_{1,3} = ⟨⟨c)(a)⟩⟩ and P_{2,3} = ⟨⟨c)(a)⟩⟩ be two patterns with two different gap constraints and let us consider the sequences of Table 1. Thus, P_{1,3} matches sequence 1 (that has one word between (c) and (a)) and sequence 4 (that has no word between (c) and (a)) whereas P_{2,3} only matches sequence 1.

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

2.3 Emerging Patterns

Emerging patterns are defined as sequential patterns whose support increases significantly from one dataset to another one. 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) ≥ ρ, with ρ > 1 and GrowthRate(P) being defined by equation 2:

\[
\text{GrowthRate}(P) = \begin{cases} 
\infty, & \text{if sup}_{D_1}(P) \neq 0 \text{ and sup}_{D_2}(P) = 0 \\
\frac{\text{sup}_{D_1}(P)}{\text{sup}_{D_2}(P)}, & \text{otherwise}
\end{cases}
\]

(2)

with sup_{D_1}(P) (respectively, sup_{D_2}(P)) being the relative support of the pattern P in D_1 (respectively, in D_2).
In the paper, the datasets correspond to sets of frequent sequential patterns extracted during a mining process (as presented in Section 2.2); each set contains the sequential patterns corresponding to one corpus. The emerging patterns are then selected (according to Equation 2) from each sequential pattern dataset to the other datasets, the other datasets being considered as a whole.

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 corresponding to the considered type of text; the texts are provided by the French resources of the CNRTL\(^1\). For example, authors from poetry include Lamartine or Musset, whereas Hugo and Lamennais are part of the authors of letters, and Chateaubriand and Zola are examples of authors from 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 following set: \{.,,!,?;\}. Table 2 gives some details on these pre-processed corpora: their number of authors, works, sequences, and words.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>#authors</th>
<th>#works</th>
<th>#sequences</th>
<th>#words</th>
</tr>
</thead>
<tbody>
<tr>
<td>poetry</td>
<td>27</td>
<td>48</td>
<td>131 116</td>
<td>1 167 422</td>
</tr>
<tr>
<td>letters</td>
<td>5</td>
<td>9</td>
<td>234 997</td>
<td>1 362 543</td>
</tr>
<tr>
<td>fiction</td>
<td>37</td>
<td>9</td>
<td>663 860</td>
<td>5 105 240</td>
</tr>
</tbody>
</table>

After being pre-processed, the corpora were then POS tagged using Cordial\(^2\), a tagger that was shown to outperform TreeTagger on French texts. Thus, each word of the corpora was associated with its form, its lemma and its POS tag. Finally, the POS tags given by Cordial were post-processed in order to reduce

\(^1\) Centre National des Ressources Textuelles et Linguistiques : www.cnrtl.fr
\(^2\) The Cordial tagger is developed by Synapse Développement (www.synapse-fr.com)
their number and hence the number of extracted itemset patterns. First, too specific categories were merged into more general categories. 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 subcategories: adjectives, determiners, common nouns, proper nouns, demonstrative pronouns, relative pronouns, indefinite pronouns, and past participles. Then, categories corresponding to personal pronouns were decomposed into 2 tags: the personal pronoun tag (PPER), and the person (e.g. 1S for the singular first person). Moreover, categories corresponding to verbs were decomposed into 3 tags: the verb tag (V), the mode of the verb (e.g. INDP for the present of the indicative mode), and 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.

Mining Single-Item Sequences First, we considered single-item sequences of words. To perform the mining task on the three corpora, we used dm4 [12] that allows the definition of various constraints on the extracted single-item sequential patterns: their length, their frequency (by setting the threshold support minsup), or the gaps in them (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 0.001 % relative threshold corresponding 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): \([1, 1]\), \([1, 2]\), \([1, 3]\), and \([1, 5]\). Note that the gap constraint \([1, 1]\) corresponds to having no extra word between every two words of the patterns and hence to considering n-gram patterns.

Mining Itemset Sequences Finally, we considered itemset sequences, where each itemset represents a word with its form, its lemma, and its POS tag (thus, a word form, a lemma, and a POS tag are the items composing each itemset). For example, the itemset corresponding to the word papers is (papers paper NC). To mine these itemset sequences, we chose CloSpan [9] 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%. Indeed, because no gap constraints 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 $\rho$ to 1.001, to make it just above 1. 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 extracted with data mining techniques. These results allow us to set constraints on the sequential 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. Note that the set of patterns with a $[1, 3]$ gap constraint contains the set of patterns with a $[1, 1]$ gap constraint as well as the set of patterns with a $[1, 2]$ gap constraint. Moreover, the ratio of emerging patterns is also given for each type of patterns. We can see that a lot more itemset patterns are extracted as compared to single-item patterns. Furthermore, the ratio of emerging patterns is different for the various corpora, especially when comparing letters and fiction. Indeed, the ratio for letters is ten times higher than the one for fiction (except when considering itemset patterns where the ratio for fiction is the highest). Finally, we can see that selecting emerging 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 stylitics.

Table 3. Number of patterns for the various corpora (and ratio of emerging patterns)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Single-item patterns with gaps</th>
<th>Itemset patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[1, 1]</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>poetry</td>
<td>18816</td>
<td>37933</td>
</tr>
<tr>
<td>letters</td>
<td>16936</td>
<td>36849</td>
</tr>
<tr>
<td>fiction</td>
<td>78210</td>
<td>175645</td>
</tr>
<tr>
<td>Total</td>
<td>113962</td>
<td>230427</td>
</tr>
</tbody>
</table>

Then, we study the distribution of emerging patterns across their length. Figure 2 plots the relative number of patterns for the various pattern lengths, for the single-item patterns (with the previous various gap constraints) and for the itemset patterns, by considering the three corpora as a whole. We can see that itemset patterns are longer than single-item patterns. Indeed, the length of single-item patterns is more between 2 and 5 whereas the length of itemset
Fig. 2. Distribution of emerging patterns across lengths

patterns is more between 5 and 11. Therefore, itemset patterns are interesting
to identify longer linguistic patterns.

Finally, we study the distribution of emerging patterns across 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. 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
rate of emerging patterns is stable for growth rates greater than 10. It means
that most of the emerging patterns only appears in a certain type of text and
not at all in the other types of text.

From these quantitative results, constraints can now be set in order to nar-
row down the number of sequential patterns to analyze. Therefore, only emerging
single-item patterns and emerging itemset patterns fulfilling the following con-
straints will be analyzed from a linguistic point of view in Section 3.3:

- the pattern length is greater than 3;
- the growth rate is equal to \( \infty \).

In addition, for single-item patterns, the gap constraint is set to \([1, 3]\) (trade-off
between the number of sequential patterns and their relevance). And, for itemset
patterns, only patterns containing both POS tags and word forms or lemmas are
considered (patterns with only POS tags or with only word forms or lemmas are
discarded).

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.
Fig. 3. Distribution of emerging patterns across growth rates

First, we consider emerging 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. Furthermore, we also illustrate each pattern with examples of underlying sequences in poetry. It can be seen that single-item patterns have a more general dimension than n-grams, thanks to the gap constraint used during the pattern mining process. Thus, these patterns allow the observation of schematic grammatical structures that are relatively lexicon-independent. In fact, single-item patterns do not really allow the direct observation of stylistic specificities of a certain type of text. However, they represent fundamental clues from which the linguists can extract regular lexical-grammatical patterns that correspond to actual relevant units representative of a certain type of text.

Table 5 gives the correspondence between the single-item patterns presented in Table 4 and their associated grammatical patterns. In fact, these grammatical patterns constitute a subset of the database sequences represented by the single-item patterns. Indeed, a single-item sequence may correspond to several grammatical patterns (as shown in Table 5). Note that the grammatical patterns actually correspond to itemset patterns. Thus, grammatical patterns can be directly observed from the emerging itemset patterns. In fact, the grammatical patterns considered here correspond to collocational frameworks to the sense of Renouf and Sinclair [1], i.e., collocations on grammatical units and not on lexical units. This notion from corpus linguistics represents discontinuous sequences of two grammatical words enclosing a lexical word. An example of such a collocational framework is “a ? of” (represented by “a lot/number/kind/matter/... of”), where the symbol ? stands for variable lexemes that share common characteristics and hence form homogeneous classes. Since Renouf and Sinclair paper,
Table 4. Examples of characteristic single-item patterns from poetry

<table>
<thead>
<tr>
<th>Single-item pattern</th>
<th>Example (with English translation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>des plus que</td>
<td>il a des morsures plus venimeuses que celles de ta bouche</td>
</tr>
<tr>
<td>(some more than)</td>
<td>(he has some bites more venomous than those from your mouth)</td>
</tr>
<tr>
<td>on et on</td>
<td>une rose qu'on respire et qu'on jette</td>
</tr>
<tr>
<td>(we and we)</td>
<td>(a rose that we smell and that we throw)</td>
</tr>
<tr>
<td>le la l' que qui</td>
<td>la nuit qui m'opprime et qui trouble mes yeux</td>
</tr>
<tr>
<td>(the that and that)</td>
<td>(the night that oppresses me and that troubles my eyes)</td>
</tr>
<tr>
<td>le du qui dans</td>
<td>le gret qui resonne et le troupeau qui bele</td>
</tr>
<tr>
<td>(the of the that in)</td>
<td>(the bell that resounds and the flock that blats)</td>
</tr>
<tr>
<td>est un qui</td>
<td>est-ce un goeland qui bat de l'aile ?</td>
</tr>
<tr>
<td>(is a that)</td>
<td>(is it a pull that flaps its wing?)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Single-item pattern</th>
<th>(English translation)</th>
<th>Grammatical pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>des plus que</td>
<td>(some more than)</td>
<td>some N more ADJ than</td>
</tr>
<tr>
<td>on et on</td>
<td>(we and we)</td>
<td>N that we V and that we V</td>
</tr>
<tr>
<td>le la l' que qui</td>
<td>(the that and)</td>
<td>the N that V and (that) V</td>
</tr>
<tr>
<td>le du qui dans</td>
<td>(the of the that in)</td>
<td>the N of the N that V in the N</td>
</tr>
<tr>
<td>est un qui</td>
<td>(is a that)</td>
<td>is it a N that V is like a N that V</td>
</tr>
</tbody>
</table>
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 insights when associated to an actual usage theory considering that grammatical forms come from a linguistic usage (i.e. corpus-driven approaches, in the sense of Biber [8]) 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 current data mining techniques 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, \( \text{minsup} \), on the pattern supports, it would be interesting to set a maximum threshold, \( \text{maxsup} \), on the pattern supports as well. Indeed, most interesting sequential patterns generally appear in few sequences. Thus, by discarding too frequent sequential patterns, the total number of patterns would be reduced (for instance, by setting \( \text{maxsup} = 50, 21.2 \% \) of the poetry single-item patterns would not be extracted). Moreover, it would allow us to set \( \text{minsup} \) to a lower value, and hence to generate 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 explore 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. To do so, 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. Lastly, we have discussed the improvements that could be brought to data mining techniques both by limiting the total number of extracted sequential patterns and hence to analyze (by defining new constraints), and by making it easier to linguists to explore and analyze sequential patterns (by developing suitable tools for this task). Therefore, these discussions give us further leads to investigate in future studies. Some of this works is already in progress.

References