Combining sequence and itemset mining to discover named entities in biomedical texts: a new type of pattern

Marc Plantevit and Thierry Charnois

Université de Caen Basse Normandie,
CNRS, UMR6072, GREYC F-14032, France
Fax: +33231567330
E-mail: marc.plantevit@info.incaen.fr
E-mail: thierry.charnois@info.incaen.fr

Jiří Kléma

Faculty of Electrical Engineering,
Czech Technical University,
Technická 2, Prague 6, 166 27, Czech Republic
E-mail: klema@labe.felk.cvut.cz

Christophe Rigotti

Université de Lyon, CNRS,
INSA-Lyon, LIRIS, UMR5205, F-69621, France
E-mail: christophe.rigotti@insa-lyon.fr

Bruno Crémilleux*

Université de Caen Basse Normandie,
CNRS, UMR6072, GREYC F-14032, France
Fax: +33231567330
E-mail: bruno.cremilleux@info.unicaen.fr
*Corresponding author

Abstract: Biomedical named entity recognition (NER) is a challenging problem. In this paper, we show that mining techniques, such as sequential pattern mining and sequential rule mining, can be useful to tackle this problem but present some limitations. We demonstrate and analyse these limitations and introduce a new kind of pattern called LSR pattern that offers an excellent trade-off between the high precision of sequential rules and the high recall of sequential patterns. We formalise the LSR pattern mining problem first. Then we show how LSR patterns enable us to successfully tackle biomedical NER problems. We report experiments carried out on real datasets that underline the relevance of our proposition.

Keywords: LSR patterns; sequential patterns; biomedical named entity recognition problem; NER; constraint-based pattern mining.

Reference to this paper should be made as follows: Plantevit, M., Charnois, T., Kléma, J., Rigotti, C. and Crémilleux, B. (2009) 'Combining sequence and itemset mining to discover named entities in biomedical texts: a new type of pattern', *Int. J. Data Mining, Modelling and Management*, Vol. 1, No. 2, pp.119–148.

Biographical notes: Marc Plantevit is a Postdoctoral Researcher in Computer Science at GREYC Laboratory (CNRS UMR 6072), University of Caen, France. He is currently working on the ANR (French National Research Agency) funded project Bingo2 ANR-07-MDCO-014. He received his PhD in Computer Science in 2008 from the University of Montpellier, France, in the field of sequential pattern mining. His main research interests include sequential pattern mining, text mining and knowledge discovery in multidimensional databases.

Thierry Charnois is an Assistant Professor at IUT Caen and at the GREYC Laboratory (CNRS UMR 6072), University of Caen, France. He holds a PhD in Computer Science (1999) from LIPN Laboratory, University of Paris 13, in the field of natural language processing (NLP). His research interests include NLP and semantics, discovery knowledge and information extraction from texts, and its applications to the biomedical domain.

Jiří Kléma received his PhD in Artificial Intelligence and Biocybernetics from the Czech Technical University (CTU) in Prague in 2002. In 2005–2006, he carried out Post-Doctoral training at the University of Caen, France. Currently, he is an Assistant Professor at CTU. His main research interest is data mining and its applications in industry, medicine and bioinformatics. He focuses namely on knowledge discovery and learning in domains with heterogeneous and complex background knowledge. He is a co-author of 15 journal publications and book chapters, a Reviewer for several international journals and a member of the Presidium of The Czech Society for Cybernetics and Informatics.

Christophe Rigotti is an Assistant Professor at INSA-Lyon (University of Lyon) and works at the LIRIS laboratory (UMR5205 CNRS). He holds a PhD in Computer Science (1996) from INSA-Lyon/University of Lyon, in the field of object-oriented and deductive databases. Since then, he has been working on multi-dimensional databases, constraint programming and data mining. In data mining, his main research interests include sequential pattern mining and condensed representations for pattern extraction. He is a co-author of over 40 papers in journal and conference proceedings and he is member of many international committees.

Bruno Crémilleux is a Professor in Computer Science at GREYC Laboratory (CNRS UMR 6072), University of Caen, France. He is currently heading the data mining research group. He received his PhD in 1991 from the University of Grenoble, France and a Habilitation degree in 2004 from the University of Caen. His current research interests include knowledge discovery in databases, machine learning, text mining and their applications notably in bioinformatics and medicine. He is a co-author of over 50 papers in journals and conference proceedings and he is a member of many international conference committees.

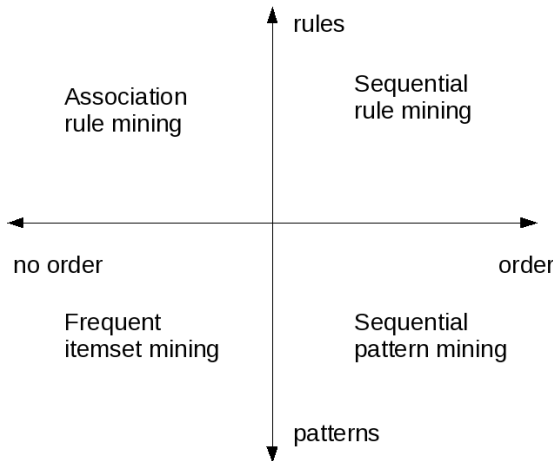
1 Introduction

In current scientific, industrial or business areas, one of the critical needs is to derive knowledge from huge datasets or text collections. This task is at the core of the knowledge discovery in database (KDD) area. In particular, a large part of the biological information is available in natural language in research publications, technical reports, websites and other text documents. A critical challenge is then to extract relevant and useful knowledge dispersed in such text collections. Lots of efforts have been made such as designing efficient tools to tackle large datasets and the discovery of patterns (i.e., relationships in the data) of potential interest to the user. Text mining in general and information extraction (IE) in particular are rapidly becoming an essential component of various bio-applications. These techniques and natural language processing (NLP) have been widely applied to extract and exploit background knowledge from biomedical texts. Among many tasks, a crucial issue is the annotation of a large amount of genetic information. IE and NLP aim at processing accurate parsing to extract specific knowledge such as named entities (e.g., gene, protein) and relationships between the recognised entities (e.g., gene-gene interactions, biological functions). The need of linguistic resources (biological databases, ontologies and IE rules such as grammars or patterns) is a common feature of the methods provided by the literature. Difficulties are well-known: multi-sense words, no formal criterion, multi-word terms and variations in gene/protein names. These linguistic issues are often handled using rules. But, except very few attempts (Califf and Mooney, 1999; Smith et al., 2008), such rules are manually elaborated and texts, which can be processed are necessarily specific and limited. Furthermore, machine learning (ML) based methods such as support vector machines, conditional random fields, etc., (Smith et al., 2008) need many features and their outcomes are not really understandable by a user. In this case, using them is not satisfactory. Indeed, we are interested in discovering knowledge, which can be easily managed and used in NLP systems in the form of linguistic patterns or rules. One of the strengths is the ability to judge, modify, enhance and improve patterns by a linguistic expert. Although this point is not further addressed here, ultimate understandability makes an important feature of the proposed methodology. Moreover, the method can be straightforwardly applied to any domain without additional effort to develop custom features or hand-crafted rules.

In this paper, we focus on the automated recognition of named entities in general and gene and protein names in particular. Even though this problem has already been tackled by a great range of various methods (see Section 5), it still remains a challenging research issue. We experimentally prove that the pattern mining approach is able to distinguish subtle relationships in text collections to highlight named entities. Sequential patterns [also referred to as sequences in Srikant and Agrawal (1996)] and sequential rules [also referred to as episode rules in Mannila et al. (1997)] are the basis of pattern mining techniques from texts because they take into account the order between the elements of texts. For text entity recognition, the experiments carried out in Section 2 show that sequences can provide suitable scores in recall whereas sequential rules show higher precision. Using only sequential patterns or only sequential rules are not enough to get sufficient recall and precision scores. Our key idea in this paper is to take benefit from synergic action of pattern and rule mining techniques. Patterns can hit a large spectrum of potentially interesting phrases while rules bring necessary precision. This synergy is

further reinforced by simultaneous application of itemset and sequential mining. Although texts must primarily be treated as sequences (of words) otherwise a large portion of information is lost, a pertinent disregard for word order can clarify the context of the core sequence. Figure 1 organises the four affined mining tasks. Despite their common grounds, we are not aware of any other work that combines their strengths in the way we do.

Figure 1 Sequential and non-sequential mining tasks



We propose a generic approach to automatically discover IE rules for the named entity recognition (NER) problem. Our main contribution is to define a method to automatically derive suitable patterns recognising gene and protein names. For that purpose, we have designed a new kind of patterns, left-sequence-right (LSR) patterns taking into account the surrounding context of a sequence and relaxing the order constraint around the sequence. These patterns provide a way to contextualise and model the neighbourhood around a sequence. They exploit the main strength of both sequences and sequential rules. Our approach is entirely automatic so that various texts, including their updates, can be handled. Furthermore, it can be applied to raw text and the discovered rules can easily be understood by the end-users.

2 Motivating example

Biomedical NER aims at identifying the boundary of a substring and then mapping the substring to a predefined category (e.g., gene or disease). Having a training corpus in which named entities are tagged, our goal is to automatically learn extraction rules that can then be applied to untagged text in order to discover named entities.

Table 1 is an example of tagged sentences that we examine in order to discover extraction rules. In these sentences, named entities are tagged in bold with surrounding ⟨…⟩. In this example, we focus on the discovery of gene names. In this section, we show that using pattern mining techniques is promising to automatically discover extraction rules of gene names.

Table 1 An example of tagged sentences from BioCreative corpus

s_1	Comparisons of the four operon control regions studied indicate that the ⟨**NarL heptamers**⟩ are arranged with diverse orientations and spacing.
s_2	Hydroxypropyl methacrylate, a new water-miscible embedding medium for electron microscopy.
s_3	⟨**Tctex-1**⟩ binding required the first 19 amino acids of Fyn and integrity of two lysine residues within this sequence that were previously shown to be important for Fyn interactions with the immunoreceptor tyrosine-based ⟨**activation motifs**⟩ of ⟨**lymphocyte Ag receptors**⟩.
s_4	Closure of an open high below-knee guillotine amputation wound using a skin-stretching device.

Prior to pattern mining application, linguistic preprocessing tasks must be carried out. The corpus has to be tokenised and then it can be stemmed. There are works devoted to this issue such as Schmid (1994). In this paper, we do not focus on the preprocessing of the corpus and we use corpus sentences that are already tokenised. All substrings that are tagged as gene names are labelled with a unique label *AGENE* as shown in Table 2.

Table 2 Transformed sentences to support pattern mining techniques

s_1	Comparisons of the four operon control regions studied indicate that the AGENE are arranged with diverse orientations and spacing.
s_2	Hydroxypropyl methacrylate, a new water-miscible embedding medium for electron microscopy.
s_3	AGENE binding required the first 19 amino acids of Fyn and integrity of two lysine residues within this sequence that were previously shown to be important for Fyn interactions with the immunoreceptor tyrosine-based AGENE of AGENE.
s_4	Closure of an open high below-knee guillotine amputation wound using a skin-stretching device.

It should be noticed that the order of the words within the sentence is primordial in the NER problem since we want to discover boundaries that delimit named entities. The order relation that we considered is the order of the tokens within the sentences. A text sentence is thus seen as a sequence of tokens or stemmas.

The discovery of association rules cannot be straightforwardly applied in this problem because such rules do not take order relation into account. That is why we use sequence-based pattern mining techniques. As preliminary experiments, we applied two pattern mining techniques:

- *Sequential pattern mining* to discover sequences that contain at least one token *AGENE* and that frequently occur in the data with respect to a frequency constraint called *minsup* (minimum support threshold, where the support is simply the number of sentences in which the patterns appear). We can then try to match these specific patterns to the text sentences in order to discover gene names. As an example, the discovered sequence $\langle w_1, w_2, AGENE, w_3, w_4 \rangle$ can then be applied in texts. If $\langle w_1, w_2 \rangle$ and $\langle w_3, w_{4,} \rangle$ are matched in a sentence, then the piece of sentence between w_1, w_2 and w_3, w_4 is tagged as a gene name.

- *Sequential rule mining* considering an additional constraint called confidence threshold that enables us to discover implications (rules) between elements within the sequences. Thus, discovered rules must satisfy both conditions: minimum support threshold and minimum confidence threshold. Confidence of a rule $X \rightarrow Y$ can be interpreted as an estimate of the probability P($Y \mid X$), the probability that a sentence, containing X contains also Y after X. The confidence threshold draws the difference between sequential patterns and sequential rules. Indeed, a sequential pattern is a frequent pattern but no interrelation between elements of the sequence is measured. As an example, the sequence *the AGENE* can be frequent on the dataset. However, there is no implication between *the* and *AGENE*. Indeed, many words different from *AGENE* appear after *the* in texts. So, the confidence threshold is not likely to be satisfied for the rule *the* → *AGENE*. While if *AGENE* appears nearly all the times after the sequence of words *the overexpression of*, then we could expect to have the rule *the overexpression of* → *AGENE* satisfying the confidence threshold. In the NER problem, the purpose is to discover rules where the if-part is a sequence of tokens and the then-part is the special token *AGENE*. These rules enable us to identify the left context of a gene name. By inverting the order relation, other rules can be inferred and the right context can also be identified. Then a pair of rules can be applied to detect the presence of a named entity and then to define its left and right boundaries. For instance, $R_l = \langle w_1, w_2, w_3 \rangle \rightarrow A\,GENE$ and $R_r = A\,GENE \leftarrow \langle w'_1, w'_2, w'_3 \rangle$ can be matched to the sentence $...w_1\,w_2\,w_3\,XYZ\,w'_3\,w'_2\,w'_1...$ where *XYZ* is then tagged as a gene name.

In order to define extraction rules that can be applied in text, we put some time constraints on the sequential patterns and sequential rules that we want to mine. Indeed, we want to discover frequent sequences of contiguous words to use the discovered patterns and rules as regular expressions in text.

To measure the relevancy of sequential patterns and sequential rules for the NER problem, we performed experiments on three different datasets. We used two well-known corpora from the literature that have frequently been used as benchmark in several papers and challenges: *GeneTag* from *Genia* dataset by Tanabe et al. (2005) and *BioCreative* dataset from Yeh et al. (2005) (the best F-score for gene/protein name extraction on these corpora are respectively 77.8% and 80%). Furthermore, we consider a very large corpus to fully benefit from scalability of the proposed pattern mining techniques. This corpus, called *Abstracts*, clearly demonstrates that this work handles very large datasets. It contains a set of 35,192 abstracts (305,192 sentences, 44.2 MB of data) collected automatically from NCBI website (http://www.ncbi.nlm.nih.gov). It is a raw text in which each abstract can be seen as a paragraph, the gene and protein name occurrences have been automatically annotated. 228,985 sentences contain at least one gene/protein name. The annotation process is a straightforward projection of terms from a dictionary, which has been learned by Charnois et al. (2006).

We separately applied sequential pattern and rule mining techniques to recognise gene and protein names in these three corpora. For the evaluation, we used a ten-fold cross-validation to partition each initial dataset in a training set and in a testing set. Unfortunately, these techniques did not lead to good results for NER problems as the following experiments show:

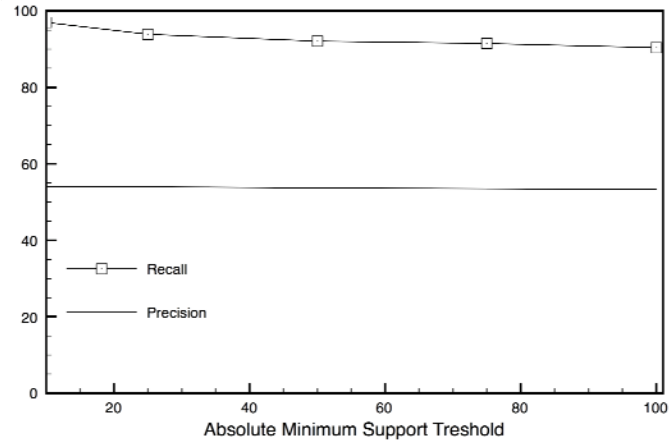
- Graphs from Figures 2(a), 3(a) and 4(a) report the recall and the precision of sequential patterns for gene recognition on the different datasets. We can see that the sequential pattern technique provides very good recall results. Sentences that contain a named entity are widely covered by these patterns. However, these patterns suffer from a lack of precision. Indeed, they cause too many false positives. In numerous cases, sequential patterns match with sentences that do not contain a named entity and then unfortunately identify a word or a group of words as a named entity. As an example, the sequential pattern ⟨*AGENE expression*⟩ enables us the discovery of many gene names but it also engenders the detection of false positives like '*this gene*' or '*the*' in sentences containing '*this gene expression*' or '*the expression*'.
- Graphs from Figures 2(b), 3(b) and 4(b) report the recall and the precision of sequential rules for gene recognition on the different datasets. These curves show that the sequential rule technique provides a good precision by virtue of the confidence measure but the recall is too low. Indeed, discovered sequential rules do not correctly cover sentences that contain a named entity in Figures 2(b) and 3(b). It is due to the fact that many rules are not taken into account because they do not respect the confidence threshold. Note that the precision in Figure 3(b) is not defined when absolute support threshold is set to 50; indeed, the recall is equal to 0% in this case. The third corpus *Abstracts* [Figure 4(b)] shows a different behaviour as it is automatically annotated and the annotation is known to capture the regular gene name occurrences while irregular ones might be omitted. Consequently, the number of false negatives is likely higher than the two other corpora.

It should be noticed that the precision rate seems to stay stable when the support threshold changes except when recall becomes equal to 0% as in Figure 3(b). It is explained by the definition of the precision rate $\left(P_r = \frac{TP}{TP + FP}\right)$: when the support threshold becomes lower, the recall rates increase. So, the number of true positives increases but the number of false positives increases as well. As a consequence, the ratio $\frac{TP}{TP + FP}$ cannot be straightforwardly altered by changes of the support threshold.

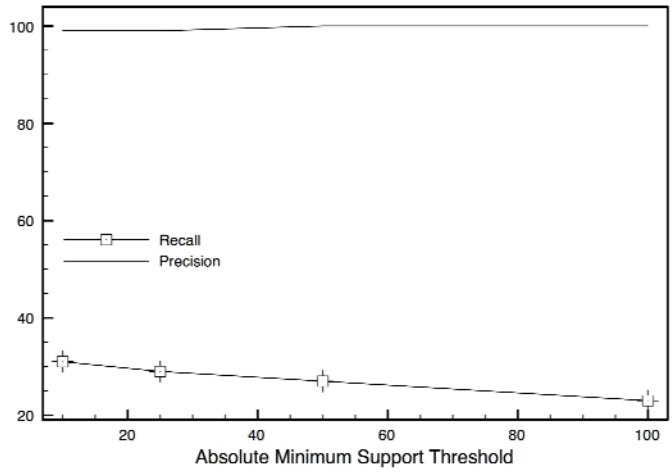
These experiments clearly show the limitations of frequent pattern mining technique for the problem of NER. On the one hand, sequential patterns offer a good coverage of sentences that contain a named entity (high recall) but lead to the detection of too many false positives (low precision). On the other hand, sequential rules provide high precision scores but too low recall. It would be very interesting to make a trade-off between the high precision of sequential rules and the high recall of sequential patterns and to profit from advantages from these kinds of patterns without their limitations. From this empirical study, we propose in this paper the LSR patterns that aim at characterising a sequence by its neighbourhood. LSR patterns combine sequential pattern mining and itemset mining by relaxing the order constraint around frequent sequential patterns. The surrounding context can then be used to check the relevancy of the pattern and thus, reduce the detection of false positives while taking advantage of the good coverage of sequential patterns.

In the rest of the paper, we define LSR patterns and describe how to mine such patterns. We also show how to use them in NER problems.

Figure 2 (a) sequential pattern mining (b) sequential rule mining $(minconf = 0.75)$ for the NER problem according to *Genia* dataset

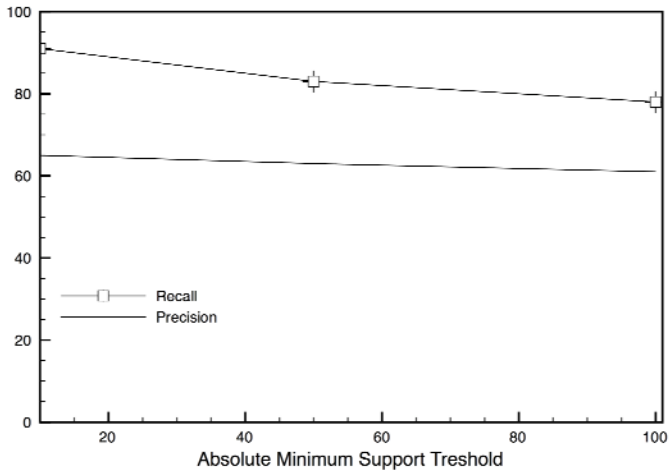


(a)

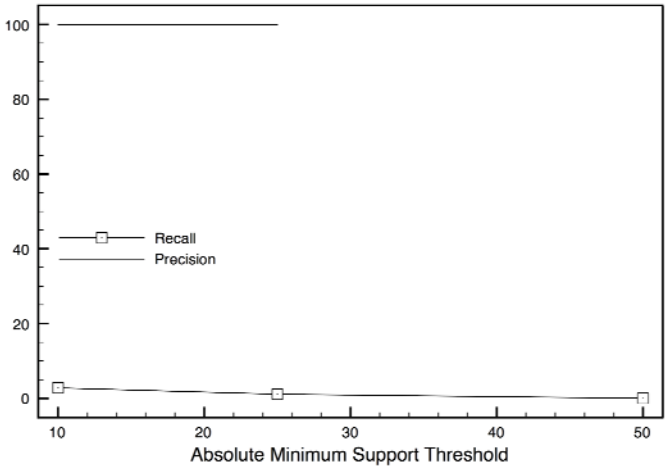


(b)

Figure 3 (a) sequential pattern mining (b) sequential rule mining $(minconf = 0.75)$ for the NER problem according to *BioCreative* dataset

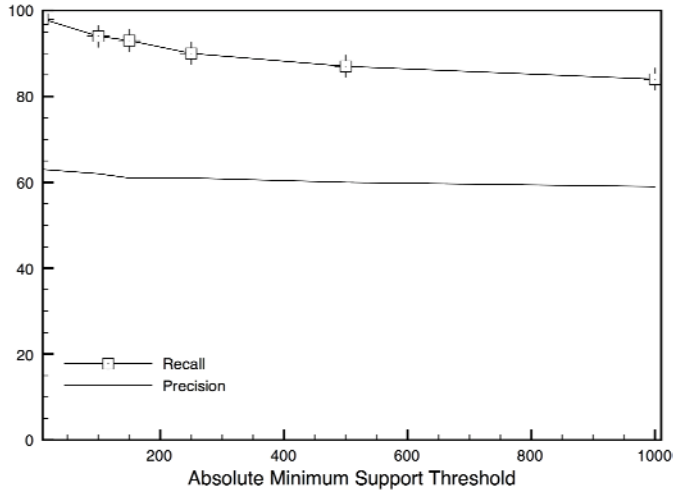


(a)

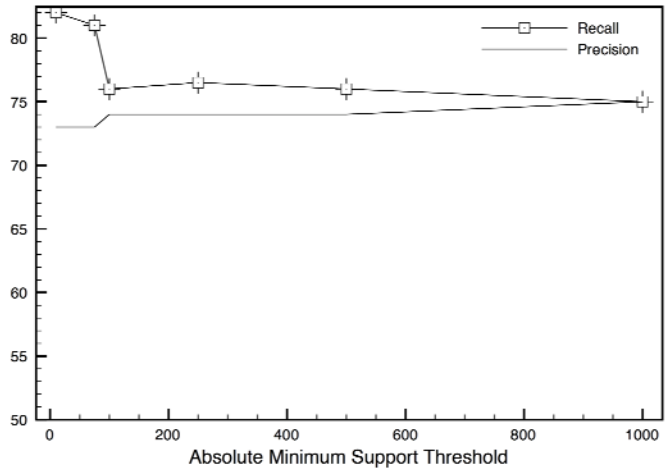


(b)

Figure 4 (a) sequential pattern mining (b) sequential rule mining $(minconf = 0.75)$ for the NER problem according to *Abstracts* dataset



(a)



(b)

3 Our proposal: LSR patterns

Before defining LSR patterns, their extraction and their application to NER problems, let us introduce some preliminary concepts related to sequences and constraints.

3.1 Preliminary definitions

Let $\mathcal{I} = \{e_1, e_2, ..., e_n\}$ be a set of *items*. A *sequence* $s = \langle i_1, i_2, ..., i_k \rangle$ is an ordered list of items. A *sequential pattern* is simply a sequence. A *data sequence* S is a sequence with

time stamps associated to each of its elements. More precisely, a data sequence S is a list $\langle(t_1, j_1),(t_2, j_2),...,(t_m, j_m)\rangle$ where $t_1 < t_2 < ... < t_m$ are time stamps and $j_1, j_2, ..., j_m$ are items. Given a sequential pattern $s = \langle i_1, i_2, ..., i_k \rangle$, a data sequence $o = \langle(u_1, i_1),(u_2, i_2),...,(u_k, i_k)\rangle$ is an occurrence of s in a data sequence S if all elements of o are in S. For instance, $\langle(1,a),(4,b)\rangle$ is an occurrence of the sequential pattern $s = \langle a,b\rangle$ in data sequence $S = \langle(1,a),(2,c),(4,b),(6,b)\rangle$.

A sequence database SDB is a set of tuples (sid, S) where sid is a sequence-id and S a data sequence. A data sequence S is said to *contain* a sequential pattern s, if s has at least one occurrence in S. The *support* of a sequential pattern s in a sequence database SDB is the number of data sequences of SDB that contain s.

Given a minimum support threshold $minsup$, the goal of mining sequential patterns on a sequence database SDB is to find the complete set of sequences whose support is greater than or equal to $minsup$.

Pattern mining involves different challenges, such as designing efficient tools to tackle large datasets and to select patterns of potential interest. The constraint-based pattern mining framework is a powerful paradigm to discover new highly valuable knowledge (see Ng et al., 1998). Constraints allow user to focus on the most promising knowledge by reducing the number of extracted patterns to those of potential interest. There are now generic approaches to discover patterns and sequential patterns under constraints (e.g., De Raedt et al., 2002; Soulet and Crémilleux, 2005; Pei et al., 2002; Garofalakis et al., 1999; Leleu et al., 2003). Note that constraint-based pattern mining challenges two major problems in pattern mining: effectiveness and efficiency. Indeed, mining may lead to knowledge flooding with patterns uninteresting to users and it often takes substantial processing power for mining the complete set of patterns in large databases. So, constraints can be used to enhance both the quality of discovered patterns and the mining process.

Let constraint C for a sequential pattern s be a Boolean function $C(s)$. A set of constraints $\mathcal{C} = \{C_1, C_2, ..., C_n\}$ for a sequential pattern s is then the conjunction of all Boolean functions $C_i(s)$ from $\mathcal{C}$. Then, given a set of constraints $\mathcal{C}$, the problem of constraint-based sequential pattern mining is to find the complete set of sequential patterns satisfying every condition C_i from $\mathcal{C}$.

Note that even if the support condition is a constraint, it does not belong to $\mathcal{C}$. Indeed, pattern mining is based on this key condition and $\mathcal{C}$ models *additional constraints different from frequency constraint*. There are various types of constraints such as *syntactic*, *length*, *duration* and *gap constraints*, see Pei et al. (2002).

3.2 LSR pattern: a new kind of pattern

As we have noticed, sequences and sequential rules present some non-negligible limitations for NER problem in biomedical data. In order to take advantage of the high recall of sequences and improve their precision, we propose a new type of pattern called LSR pattern. They enable us to characterise a sequence with itemsets representing its

surrounding context. Indeed, the key idea is to relax the order constraint around a sequence in order to model left and right neighbourhood of a sequence, thanks to itemsets.

Definition 3.1: (LSR) A LSR pattern x is a triplet $x=(l,s,r)$ where:

- s is a sequential pattern
- l and r are sets of items.

LSR patterns go further than the result of a combination between a sequence and two itemsets. Indeed, such patterns provide a way to contextualise a sequence, thanks to its neighbourhood. Thus, itemsets l and r provide a way to model neighbourhood around sequence s. As an example, consider an LSR pattern $x_1=(\{\},\langle the, AGENE,\rangle\{gene, with, associated\})$ where $l=\{\}$ and $r=\{gene, with, associated\}$ which means that these words are in the right neighbourhood of the sequence *the AGENE*.

The order relation constraint is relaxed around frequent sequential patterns in data sequences in order to extract frequent itemsets that model the neighbourhood of the sequence and contextualise it in the data sequences. To formalise the extraction of frequent LSR patterns, we need to introduce the following definitions.

Contrary to an itemset that occurs at most once in a transaction in the itemset mining problem, a sequence may appear several times in a data sequence (see example below). Consequently, for a same data sequence, there are different ways to identify the neighbourhood of a sequence within the data sequence. In order to exhibit the *most representative* itemsets that model neighbourhood, we introduce the notion of 'compact *occurrence of* s *in a data sequence*'.

Definition 3.2: (compact occurrence) Given a sequential pattern $s=\langle i_1,i_2,...,i_k\rangle$, a set of constraints $\mathcal{C}$ and a data sequence S, then an occurrence o_c of s in S, where $o_c=\langle(t_1,i_1),(t_2,i_2),...,(t_k,i_k)\rangle$ is a compact occurrence of s in S if the following conditions hold:

- o_c satisfies $\mathcal{C}$
- there is no occurrence $o'=\langle(t'_1,i_1),(t'_2,i_2),...,(t'_k,i_k)\rangle$ of s in S such that $o'\neq o_c$ and o' satisfies $\mathcal{C}$ and $t_1\leq t'_1$ and $\forall\alpha\in\{2,...,k\}, t'_\alpha\leq t_\alpha$.

This definition enables to focus on the minimal pieces of the data sequence S that contain the sequence s. Indeed, a data sequence S can contain several compact occurrences of a sequence s. Compact occurrences have similar semantics to 'minimal occurrences' from Mannila et al. (1997).

As an example, given $\mathcal{C}=\emptyset$, the data sequence $S=\langle(1,a),(2,c),(3,b),(4,d),(5,a),(7,a),(8,b),(10,e),(12,f),(14,a),(15,g),(16,h),(18,b),(20,c)\rangle$ contains three different compact occurrences of $s=\langle a,b\rangle$:

1 $\langle(1,a),(3,b)\rangle$

2 $\langle(7,a),(8,b)\rangle$

3 $\langle(14,a),(18,b)\rangle$.

Note that $\langle(1,a),(18,b)\rangle$ is not a compact occurrence of s in S since it is not minimal.

If we add a maximal gap constraint $max\_gap = 2$ to $\mathcal{C}$ meaning that the maximal time gap between consecutive items of the sequence s is 2, then S contains two compact occurrences of s : $\langle(1,a),(3,b)\rangle$ and $\langle(7,a),(8,b)\rangle$.

Since a data sequence can contain several compact occurrences of s, we speak of the ith *compact occurrence* of s in S (denoted by o_c^i) where i refers to order of appearance of the compact occurrence within the data sequence. According to the previous example, where $\mathcal{C}=\emptyset, \langle(1,a),(3,b)\rangle$, $\langle(7,a),(8,b)\rangle$ and $\langle(14,a),(18,b)\rangle$ are respectively the first, second and third compact occurrences of s in S.

In order to define a way to identify neighbourhood with itemsets, we have to define the notion of the prefix of a ith compact occurrence.

Definition 3.3: (prefix of an occurrence) Let o_c^i be the ith compact occurrence of s in S, the prefix of o_c^i in S is equal to the subsequence of S starting at the beginning of S and ending strictly before the first item of o_c^i.

In our example, where $S = \langle(1,a),(2,c),(3,b),(4,d),(5,a),(7,a),(8,b),(10,e),(12,f),(14,a),(15,g),(16,h),(18,b),(20,c)\rangle$ is the input sequence, $s = \langle a,b\rangle$ and $\mathcal{C}=\emptyset$, we have:

- the prefix of the first compact occurrence o_c^1 of s in S is equal to $\langle\rangle$
- the prefix of o_c^2 is equal to $\langle(1,a),(2,c),(3,b),(4,d),(5,a)\rangle$
- the prefix of o_c^3 is equal to $\langle(1,a),(2,c),(3,b),(4,d),(5,a),(7,a),(8,b),(10,e),(12,f)\rangle$

In the same way, we introduce the notion of suffix of a ith compact occurrence in a data sequence.

Definition 3.4: (suffix of an occurrence) Let o_c^i be the ith compact occurrence of s in S, the suffix of o_c^i in S is the subsequence of S starting just after the last item of o_c^i to the end of S.

According to our example where $s = \langle a,b\rangle$ and $\mathcal{C}=\emptyset$, we have the following suffixes:

- the suffix of the first compact occurrence o_c^1 of s in S is equal to $\langle(4,d)(5,a),(7,a),(8,b),(10,e),(12,f),(14,a),(15,g),(16,h),(18,b),(20,c)\rangle$
- suffix of o_c^2 is equal to $\langle(10,e),(12,f),(14,a),(15,g),(16,h),(18,b),(20,c)\rangle$
- suffix of o_c^3 is equal to $\langle(20,c)\rangle$.

In order to delimit the range of the neighbourhood around compact occurrences, we introduce a parameter N_R to consider only items having time-stamps sufficiently close to compact occurrences (absolute difference between item time-stamp and time-stamp of the closest element of a compact occurrence must not be greater than N_R). This constraint is taken into account in the prefix and the suffix of the ith compact occurrence of s in S. Indeed, only items which respect neighbourhood range N_R to o_c^i are returned.

According to our example, given $s=\langle a,b\rangle$, $\mathcal{C}=\emptyset$ and $N_R=5$:

- $prefix\left(o_c^i,S,N_R\right)=\langle\ \rangle$ and $suffix\left(o_c^i,S,N_R\right)=\langle(4,d),(5,a),(7,a),(8,b)\rangle$
- $prefix\left(o_c^2,S,N_R\right)=\langle(2,c),(3,b),(4,d),(5,a)\rangle$ and $suffix\left(o_c^2,S,N_R\right)=\langle(10,e),(12,f)\rangle$
- $prefix\left(o_c^3,S,N_R\right)=\langle(10,e),(12,f)\rangle$ and $suffix\left(o_c^3,S,N_R\right)=\langle(20,c)\rangle$.

Note that N_R can be automatically set by studying the average size of the prefix and the suffix of compact occurrences.

Definition 3.5: (inclusion of LSR pattern) Given N_R and a set of constraints $\mathcal{C}$, a LSR pattern $x=(l,s,r)$ is included in a sequence S if the following conditions held:

1 s has a compact occurrence in S

2 $\exists i$ such that $\forall_{el}\in l$, item e_l appears in $prefix\left(o_c^i,S,N_R\right)$ and $\forall_{er}\in r$, item e_r appears in $suffix\left(o_c^i,S,N_R\right)$, where o_c^i is the ith compact occurrence of s in S.

To support a LSR (l,s,r) pattern, a data sequence first must contain the sequential pattern s. Then, it must exist o_c^i, an ith compact occurrence of s in S such that all elements of l must be contained in the prefix of o_c^i with respect to N_R. Moreover, for the same compact occurrence o_c^i, all elements of r must also be contained in the suffix of o_c^i with respect to N_R. Note that the order constraint is relaxed for l and r. Indeed, elements from these itemsets must be contained in the neighbourhood of the sequence, whatever their order of appearance.

According to the previous definition, we can define the support of a LSR pattern in a sequence database.

Definition 3.6: (Support) Given a set of data sequences SDB and a neighbourhood range N_R, the support of a LSR pattern x is the number of sequences from SDB that contain x.

The problem of mining LSR patterns aims at discovering *frequent* LSR patterns from a sequence database. In order to avoid some redundancies, we return frequent LSR patterns having maximal itemsets.

Definition 3.7: (LSR pattern mining problem) Let SDB be a set of data sequences and N_R be a radius of neighbourhood. Given a minimum support threshold $minsup$, the problem of mining LSR patterns is to find the complete set of LSR patterns FS from SDB defined as the set $FS = \{x = (l,s,r)\ s.t.\ support(x) \geq minsup$ and $\nexists x' = (l',s,r')$ having $support(x') \geq minsup$ where $l \sqsubseteq l'$ and $r \sqsubseteq r'$ and $x \neq x'\}$.

The problem of mining LSR patterns is difficult since it combines constraint based sequence mining and itemset mining when the order constrain is relaxed around a frequent sequence within data sequences. Nevertheless, the next section shows how we overcome this difficulty and it provides our method to mine LSR patterns.

3.3 LSR pattern mining algorithm

Our method to extract LSR pattern is divided into two constraint-based mining steps. First, the set $SAT(\mathcal{C})$ of sequential patterns that satisfy the set of constraints $\mathcal{C}$ is discovered from SDB. Then, a new set SDB' of data sequences is generated according to patterns from $SAT(\mathcal{C})$. The LSR patterns are then extracted from this dataset SDB'.

Algorithm 1 describes the extraction of frequent LSR patterns. Let us describe more precisely its different steps.

Given SDB, $minsup$, and $\mathcal{C}$, the first step of the algorithm is to find the set of sequential patterns in SDB that satisfy $\mathcal{C}$, denoted $SAT(\mathcal{C})$.

Then, the algorithm transforms SDB into a new sequence dataset SDB' according to $SAT(\mathcal{C})$. It builds a set of identifiers $\mathcal{P}_{id}$ associated to the patterns in $SAT(\mathcal{C})$, that will be used as additional items in the new dataset. For each occurrence o_c of the patterns in $SAT(\mathcal{C})$ (first loop), a new sequence S' is built. In such a sequence S', a pattern identifier replaces the occurrence o_c, and on the left and on the right of the occurrence only the elements within the N_R neighbourhood are conserved. As an example, given a neighbourhood $N_R = 4$, and a sequential pattern $s = \langle a,b,c \rangle$, from its compact occurrence $\langle (3,a),(6,b),(9,c) \rangle$ in the data sequence $S = \langle (1,a),(2,c),(3,a),(4,d),(6,b)\ (8,d),(9,c),(11,a),(12,d),(14,e),(18,c) \rangle$ of SDB, the algorithm generates in SDB' the sequence $S' = \langle (1,a),(2,c),(3,pattId(s)),(11,a),(12,d) \rangle$.

Algorithm 1 LSR pattern mining

Data: Sequence database SDB, minimum support threshold $minsup$, set of constraints $\mathcal{C}$, neighbourhood range N_R

Result: Set of frequent LSR patterns

begin

$SAT(\mathcal{C}) \leftarrow$ FrequentSequenceMining$(minsup, SDB, \mathcal{C})$;

$SDB' \leftarrow \emptyset$;

Associate to each pattern s in $SAT(\mathcal{C})$ a new symbol denoted $pattId(s)$;

Let $\mathcal{P}_{id}$ be the set of all these new symbols;

for each compact occurrence o_c of the patterns in $SAT(\mathcal{C})$ do

Let o_c be of the form: $\langle (t_1,i_1),(t_2,i_2),...,(t_k,i_k)\rangle$;

Let S be the data sequence where o_c, occurrence of a pattern s, has been found;

$S' \leftarrow prefix(o_c, S, N_R) \oplus \langle (t_1, pattId(s))\rangle \oplus suffix(o_c, S, N_R)$

// where $\oplus$ denotes list concatenation

$SDB' \leftarrow SDB' \cup \{S'\}$

$\mathcal{C}' \leftarrow \{$pattern must contain an element of $\mathcal{P}_{id}\}$

$SAT(\mathcal{C}') \leftarrow$ FrequentSequenceMining$(minsup, SDB', \mathcal{C}')$;

$\mathcal{R} \leftarrow \emptyset$;

for each pattern p in $SAT(\mathcal{C}')$ do

Let p be of the form: $\langle i_1, i_2, ..., i_n, id, i'_1, i'_2, ..., i'_m\rangle$ where $id \in \mathcal{P}_{id}$;

Let s be the sequential pattern such that $pattId(s) = id$;

Left $left$ be the set of the different items appearing in $i_1, i_2, ..., i_n$;

Left $right$ be the set of the different items appearing in $i'_1, i'_2, ..., i'_m$;

$\mathcal{R} \leftarrow \mathcal{R} \cup \{\langle left, s, right\rangle\}$;

Remove from $\mathcal{R}$ the LSR patterns that are not maximal;

return $\mathcal{R}$;

end

Next, from SDB', the algorithm extracts, the sequential patterns that contain an identifier of one of the patterns extracted from SDB. Then (second loop), for each pattern p obtained from SDB', the algorithm retrieves the identifier part $(id \in \mathcal{P}_{id})$ to find the corresponding sequential pattern s extracted from SDB. This pattern s forms the central part of a LSR pattern. The algorithm takes the different items on the left (resp. on the right) of the id to form the left (resp. right) part of this LSR pattern. Finally,

non-maximal patterns are removed from the resulting set $\mathcal{R}$, where $x = \langle (l, s, r) \rangle$ is non-maximal if there is another LSR pattern $x' = \langle (l', s, r') \rangle$ such that $x' \neq x$ and $l \sqsubseteq l'$ and $r \sqsubseteq r'$.

The algorithm is based on two sequence mining steps. It is complete because the sequence mining algorithm, which is used, is complete.

3.4 Use of LSR patterns for NER problems

LSR patterns can be used for biomedical NER problem. To challenge this problem, we have first to extract specific LSR frequent patterns. Then, we have to correctly use this set of LSR patterns for discovering named entities in natural language texts.

3.4.1 Extracting LSR patterns for NER problems

First, we have to mine frequent LSR patterns on a tagged and tokenised corpus with special constraints. Sequences must contain a biomedical named entity. As an example, sequences must contain an item *AGENE* in the case of gene name recognition. Moreover, a time constraint is added in order to consider only consecutive events. This constraint is primordial for the use of sequences as regular expression in the recognition phase.

To extract patterns in the experiments presented in this paper, we used our own prototype implemented in C and called *dmt4sp*. This program enables to extract patterns that encompass substring patterns, serial episodes from Mannila et al. (1997) and a limited form of sequential patterns (see Agrawal and Srikant, 1995). It performs complete extractions of the patterns in a collection of sequences, under a combination of constraints on the support and syntax of the patterns, and on the time intervals between the events. The support constraint includes both the support in number of occurrences of the patterns (as defined by Mannila et al., 1997), and the support in number of sequences containing at least one occurrence of the patterns (as defined by Agrawal and Srikant, 1995). The second kind of constraints, the syntactic constraints, includes constraints on the prefix of the patterns and on the pattern sizes (minimum and maximum size). Finally, the time interval constraints enable to set the minimum and maximum time span between events and also between the first and the last element of the patterns. The pattern enumeration method is a standard depth-first prefix-based strategy. It combines constraint checking with a management of occurrences using the occurrence list approach (see Zaki, 2000) with a virtual database projection proposed by Pei et al. (2001), and an efficient handling of multiple occurrences as Meger and Rigotti (2004) and Nanni and Rigotti (2007), under the so-called minimal occurrence semantics from Mannila et al. (1997).

We propose to associate a confidence measure to the sequential pattern s of each LSR pattern. The aim of this measure is to determine if the sequential pattern can be applied on its own or if it is necessary to study its surrounding context (itemsets l and r) to apply it. The confidence of a sequential pattern s for a entity name E is equal to the support of s divided by the support of the sequential pattern s in which the items corresponding to the entity name E (e.g., *AGENE*) have been replaced by a wild-card $*$. This sequential pattern is denoted $s[E/*]$, and definitions of support and occurrence also apply to it, with the wild-card $*$ matching any word or group of words.

Definition 3.8: (Confidence) Given a named entity E, the confidence of a sequential pattern s, containing E, is equal to:

$$Confidence_E(s) = \frac{support(s)}{support(s[E/*])}$$

This measure aims at determining if the occurrence of entity E could be related to the presence of the other items of the sequence. For instance, if the support of a sequence ⟨*the gene AGENE interacts with*⟩ is similar to the support of the sequence ⟨*the gene* ∗ *interacts with*⟩ (confidence $\simeq 1$), this means that when a sentence contains '*the gene*' and further '*interacts with*', there is a gene name between them.

Algorithm 2 Use of LSR pattern for NER problems

Data: Sentence S, LSR pattern $x = (l, s, r)$, minimum confidence threshold $minconf$, neighbourhood range N_R, minimum number of words W_{min}, named entity E

begin

for each compact occurrence o_c of $s[E/*]$ in S do

if $Confidence(s) \geq minconf$ then

Label with E the part of o_c corresponding to $*$ in $s[E/*]$;

else

if $|prefix(o_c, S, N_R) \cap l| + |suffix(o_c, S, N_R) \cap r| \geq W_{min}$ then

Label with E the part of o_c corresponding to $*$ in $s[E/*]$;

else

Do not apply s;

end

3.4.2 Detection of named entities

Algorithm 2 describes how a LSR pattern can be applied or not to a sentence. Given a sentence in natural language, this sentence is tokenised and then we try to find patterns from the set of frequent LSR patterns that can be applied to the tokenised sentence. If a sequential pattern s of a LSR pattern $x = (l, s, r)$ can be applied (all tokens of s that are different from a named entity are perfectly matched), we check the confidence of s. If the confidence is greater than a minimum confidence threshold $minconf$, then, we consider that s can be applied on its own. Otherwise, s is not confident enough to be directly applied. It is thus, necessary to examine its surrounding context. If a sufficient number of items, according to a threshold W_{min}, from l and r match the left and right contexts of s within the tokenised sentence, then the use of s is considered to be relevant, according

to the context. Notice that the sequential pattern s can be applied several times within the sentence. So it is necessary to consider all compact occurrences. Since the number of compact occurrences within a sentence is finite, Algorithm 2 terminates.

4 Experiments

We report experiments performed on real datasets described in Section 2: *BioCreative* (Yeh et al., 2005, cf. Figure 5), *Genia* (Tanabe et al., 2005, cf. Figure 6) and *Abstracts* (cf. Figure 7). These experiments aim at showing the interest of LSR patterns, especially in the biomedical the NER problem where they represent an excellent trade-off between the high-precision of sequential rules and the high recall of sequential patterns. Each corpus was tokenised. According to the previous definitions, each sentence is a data sequence. SDB is then the set of sentences from a corpus. We used a ten-fold cross validation to partition each initial data set in a training set and in a testing set.

LSR patterns excel in exploitation of formerly unconfident sequential patterns. As an example, *that AGENE* is not confident at all, but some words frequently appear in the left neighbourhood of this pattern (*indicated*, *revealed*, *demonstrate*, *evidence*) and in the right neighbourhood (*binds*, *expressed*, *activity*, *protein*, etc.). As a consequence, such unconfident sequential patterns that seem to be useless for the NER problem can be applied, thanks to their neighbourhood.

The goal of the experiments is to evaluate the quality of recognition of LSR patterns for NER problems. We also study the behaviour of LSR patterns according to the minimum support, the minimum confidence and W_{min}. In all experiments, we fix $N_R = 5$ for linguistic reasons, it is a size for which linguists consider that it makes sense to try to connect words.

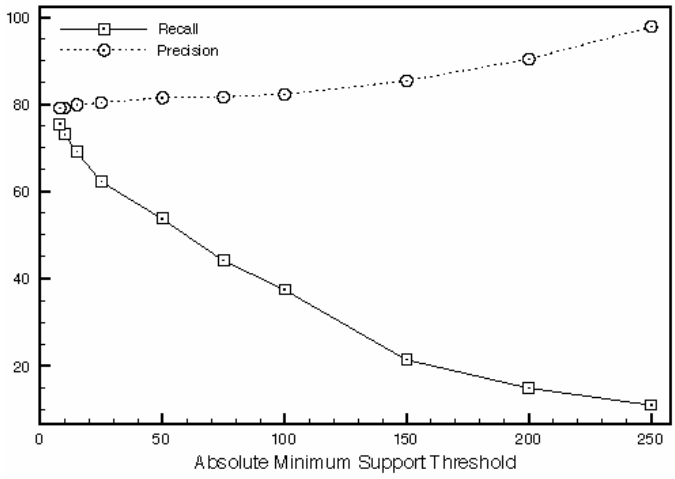
Figures 5(a), 6(a) and 7(a) describe the precision and recall of LSR patterns for NER problems according to the absolute minimum support threshold. The behaviour of LSR patterns is similar in the three plots. The recall increases and the precision decreases when the minimum support threshold becomes smaller. Indeed, there is a larger set of frequent LSR patterns that thus provides a better coverage (better recall) for the detection of named entities. However, this larger set leads also to the detection of a greater number of false positives and then to a lower precision.

Figures 5(b), 6(b) and 7(b) aim at comparing the performance of LSR patterns, sequential patterns and sequential rules for the NER problem. To compare these approaches, we use the well-known F-measure $F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$ (see Van Rijsbergen, 1979) that is the harmonic mean of precision and recall. Indeed, this measure aims to make a trade-off between precision and recall. So we use it to evaluate and compare the performance of the different approaches. Note that there is no result for sequential rules in Figure 5(b) because this technique gave too bad results in this *BioCreative* corpus [see Figure 3(b)]. For *BioCreative* corpus [Figure 5(b)], LSR patterns are significantly better than sequential patterns. On *Genia* corpus [Figure 6(b)], LSR patterns are also better when the minimum support threshold is low. On *Abstracts* corpus [Figure 7(b)], LSR patterns overall give the best F-scores. Note again that sequential rules are better than sequential patterns on this corpus. These different plots show the

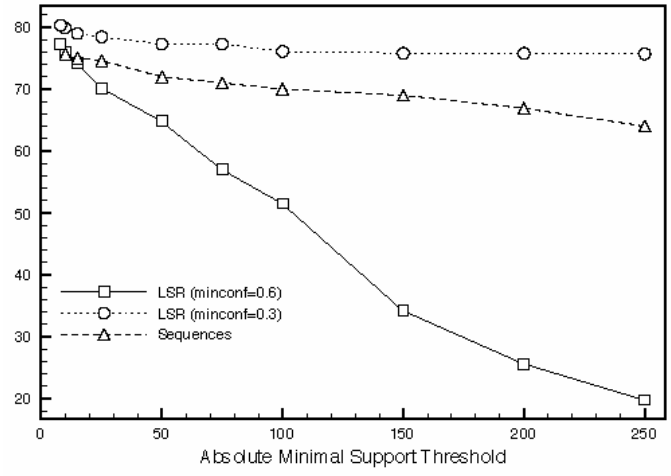
interest of LSR patterns for the NER problem since they overcome sequential patterns and sequential rules.

Figures 5(c), 6(c) and 7(c) report the recall and precision of LSR patterns when the minimum confidence threshold changes. The three plots are similar. The recall increases and the precision decreases when the minimum confidence threshold becomes lower. Indeed, the lower the confidence threshold, the bigger the number of false positives is. However, we can notice that the neighbourhood awareness lead to preserve the good precision of LSR patterns.

Figure 5 Experiment on *BioCreative* dataset, (a) precision and recall of LSR patterns $(W_{min} = 3, N_R = 5)$ (b) F-score of LSR patterns, sequential patterns $(W_{min} = 3, N_R = 5)$ (c) precision and recall of LSR patterns according to $minconf(minsupp = 10, vmin = 3, N_R = 5)$ (d) precision and recall of LSR patterns according to $W_{min}(minconf = 0.6, N_R = 5)$

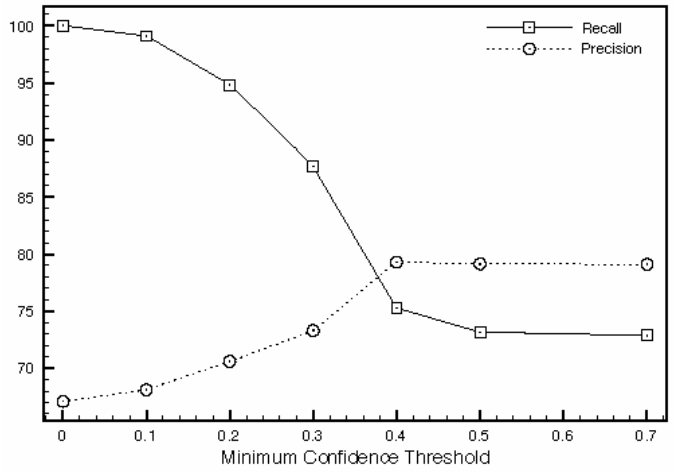


(a)

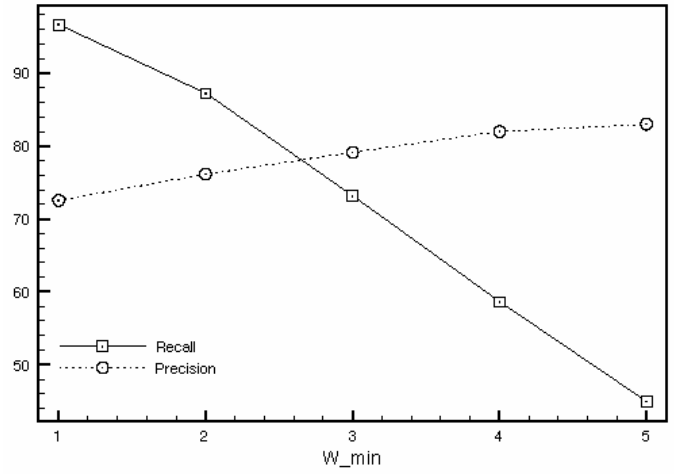


(b)

Figure 5 Experiment on *BioCreative* dataset, (a) precision and recall of LSR patterns $(W_{min}=3, N_R=5)$ (b) F-score of LSR patterns, sequential patterns $(W_{min}=3, N_R=5)$ (c) precision and recall of LSR patterns according to $minconf\,(minsupp=10, vmin=3, N_R=5)$ (d) precision and recall of LSR patterns according to $W_{min}\,(minconf=0.6, N_R=5)$ (continued)



(c)

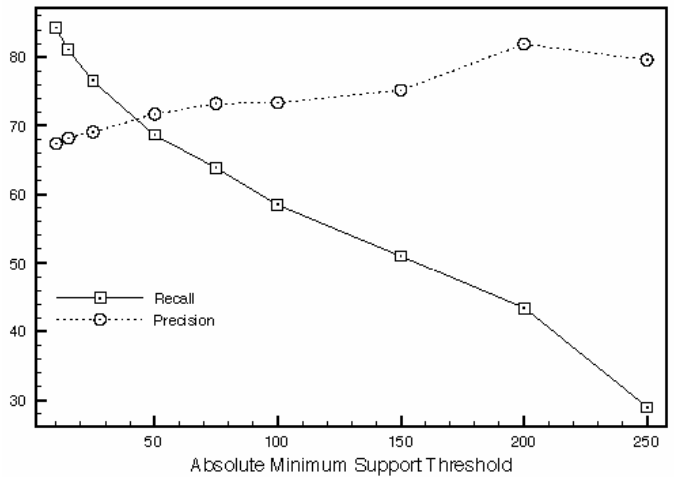


(d)

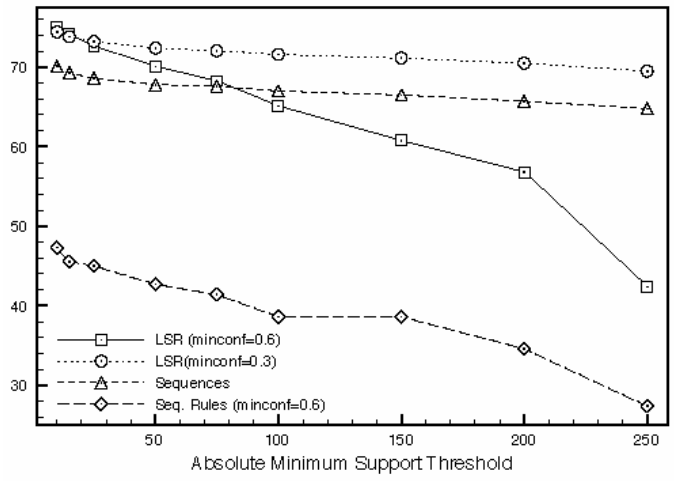
Figures 5(d), 6(d) and 7(d) report the recall and the precision of LSR patterns according to W_{min}. This parameter means that at least W_{min} items from the itemsets l and r must be present in the neighbourhood of the sequential pattern s to take the LSR pattern $x=(l,s,r)$ into account for the detection of a named entity. When W_{min} is too important, it is difficult for LSR patterns to satisfy this condition whereas they easily

satisfy it when W_{min} is small. Therefore, the precision increases and the recall decreases when W_{min} becomes higher.

Figure 6 Experiment on *Genia* dataset, (a) precision and recall of LSR patterns $(W_{min} = 3, N_R = 5)$ (b) F-score of LSR patterns, sequential patterns and sequential rules $(vmin = 3, N_R = 5)$ (c) precision and recall of LSR patterns according to $minconf(W_{min} = 3, N_R = 5)$ (d) precision and recall of LSR patterns according to $W_{min}(minsupp = 50, minconf = 0.6, N_R = 5)$

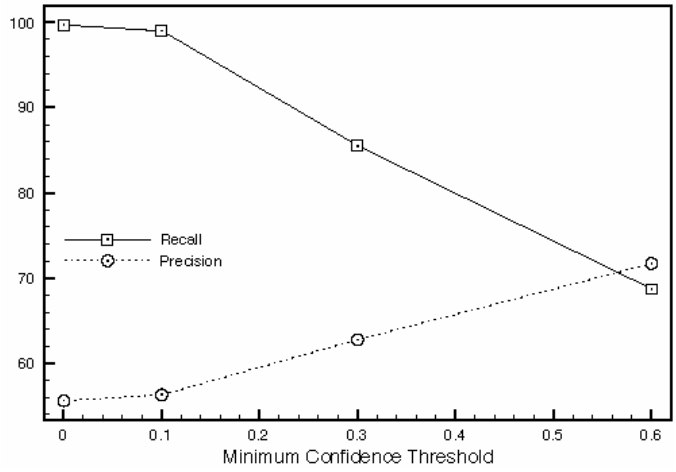


(a)

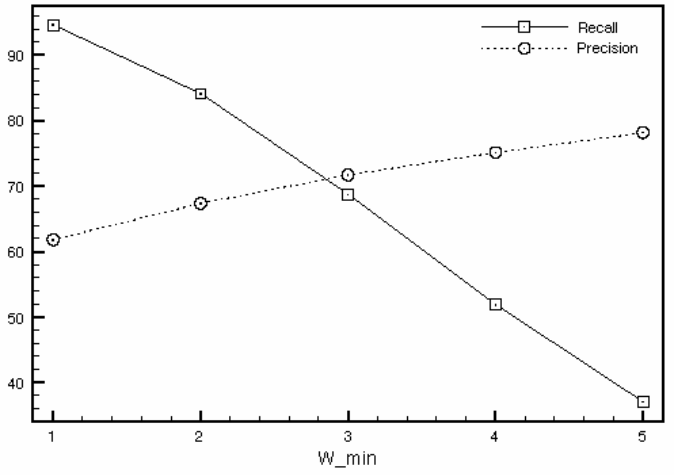


(b)

Figure 6 Experiment on *Genia* dataset, (a) precision and recall of LSR patterns $(W_{min} = 3, N_R = 5)$ (b) F-score of LSR patterns, sequential patterns and sequential rules $(vmin = 3, N_R = 5)$ (c) precision and recall of LSR patterns according to $minconf(W_{min} = 3, N_R = 5)$ (d) precision and recall of LSR patterns according to $W_{min}(minsupp = 50, minconf = 0.6, N_R = 5)$ (continued)

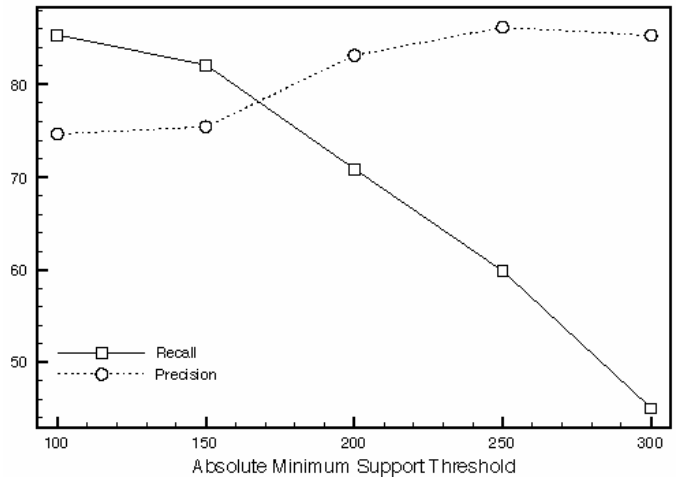


(c)

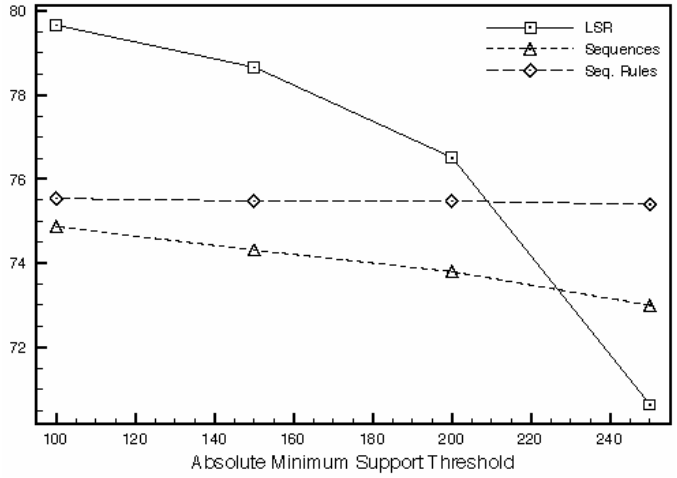


(d)

Figure 7 Experiment on *Abstracts* dataset, (a) precision and recall of LSR patterns $(W_{min}=3, N_R=5)$ (b) F-score of LSR patterns, sequential patterns and sequential rules $(W_{min}=3, N_R=5, minconf=0.6)$ (c) precision and recall of LSR patterns according to $minconf(minsupp=100, W_{min}=3, N_R=5)$ (d) precision and recall of LSR patterns according to $W_{min}(minsupp=100, minconf=0.6, N_R=5)$

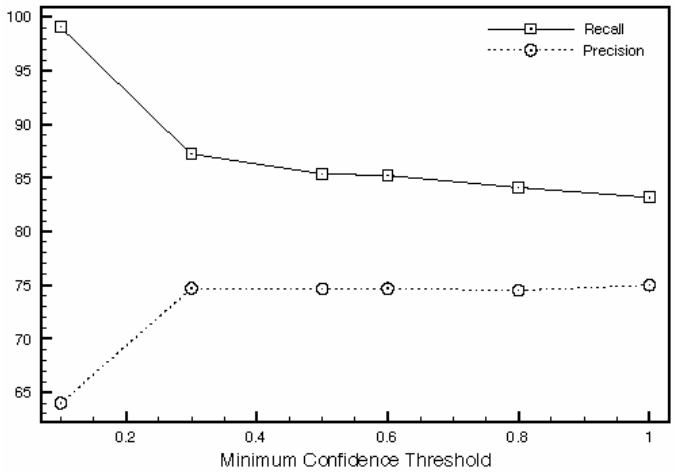


(a)

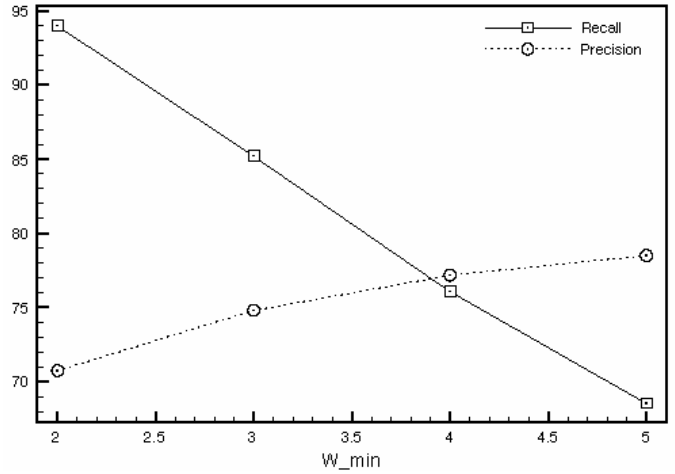


(b)

Figure 7 Experiment on *Abstracts* dataset, (a) precision and recall of LSR patterns $(W_{min} = 3, N_R = 5)$ (b) F-score of LSR patterns, sequential patterns and sequential rules $(W_{min} = 3, N_R = 5, minconf = 0.6)$ (c) precision and recall of LSR patterns according to $minconf\,(minsupp = 100, W_{min} = 3, N_R = 5)$ (d) precision and recall of LSR patterns according to $W_{min}\,(minsupp = 100, minconf = 0.6, N_R = 5)$ (continued)



(c)



(d)

These experiments show the strengths of our approach. Taking the neighbourhood of a sequential pattern into account provides promising results. Indeed, LSR patterns overcome sequential patterns and sequential rules. According to the best results for gene/protein name recognition on *Genia* and *BioCreative* corpora (F-score are respectively 77.8% and 80%), our results are comparable. Moreover, LSR patterns are easily understandable. As an example, we discover the following pattern ⟨{} ⟨*AGENE*, *expression*, *in*⟩,{*cells*}⟩ that means that the word *cells* appears in many cases in the right

neighbourhood of the frequent sequence ⟨*AGENE expression in*⟩. This illustrates another important interest of our approach: the possible use of LSR patterns in a NLP system by a linguistic expert and/or as linguistic resource.

5 Related works

NER is an IE subtask, which consists in locating some strings in a corpus and then assigning a predefined category (gene, protein, biological function) to them. NER has to deal with several difficulties such as polysemy (multi-sense words), synonymy, multi-word terms, variability in the form of names and neologism. It can be considered as a NLP problem and linguistic analysis based methods are one of the proposed approaches in literature such as Cohen and Hunter (2004) and Cohen and Hersh (2005). They are named 'rule-based approaches' since they aim at defining regular expressions, linguistic patterns or grammars that match gene or protein names [for example, Fukuda et al. (1998), one of the earliest systems]. Some of them (e.g., Humphreys et al., 2000) use terminological resources: databases, ontologies, such as UMLS and LocusLink. According to Leser and Hakenberg (2005) rule-based approaches can reach high precision but recall is often low if the rules are too specific making the system not robust enough towards new named entities. Another critical point has to be considered: the rules are manually designed by human experts, are a highly time consuming task and portability is costly.

A second broad category of approaches appeared with the availability of annotated corpus: methods based on ML techniques have been investigated and some promising results have already been obtained by cross-fertilisation of IE and ML techniques on biomedical texts (see Chang et al., 2006; Nédellec et al., 2006 for a review; Smith et al., 2008 for recent systems used during the latest BioCreative challenge, BioCreative II). A large variety of approaches can be used: decision trees, Bayesian classifiers, maximum entropy, hidden Markov models, support vector machines and conditional random fields. Some of the ML approaches use sequence based systems, considering the complete ordered sequences of words in sentences: for example Kinoshita et al. (2005) and Dingare et al. (2004). The first one retrains a dedicated train tagger (TnT-Tagger) by including sequential information, and the second one uses entropy model for predicting the most probable sequence of classifications for words of a sentence. These works often use statistical discriminators and differ from our approach by building models that can only be used as black boxes to perform predictions, but that cannot be interpreted by linguistic or biological experts. For example, SVMs draw a hyperplane in an n-dimensional space, from which deducing readable and understandable patterns is not feasible. However, there are some systems that aim at learning some linguistic rules that can be read and understood by human experts. For instance, Califf and Mooney (1999), Kim et al. (2007) and Cakmak and Özsoyoglu (2007) learned rules in the form of single slot IE patterns or textual extraction patterns, which are equivalent to our sequential patterns (the s in our LSR patterns). These systems have not been used for name gene recognition but for extracting relations between entities [relations for protein/gene annotations in Kim et al. (2007) and Cakmak and Özsoyoglu (2007)], and do not consider the intrinsic issue of high precision/low recall problem that is due to the use of sequential patterns (see Leser and Hakenberg, 2005). In our approach, this problem is overcome by the use of contextual information (the l and r parts of the LSR patterns).

Mining sequential data is not limited to the application presented in this paper and arises in many domains, to analyse various kinds of data including customer transactions, web logs, geophysical data, medical data and of course biological sequences. Most of the time, due to the size of the datasets and to the size of the pattern space, mining sequential data is a difficult task. It has received a lot of attention in the literature, from the extraction of substrings (e.g., Ukkonen, 1995) to the extraction of more general patterns like sequential patterns (e.g., Agrawal and Srikant, 1995) and episodes (e.g., Mannila et al., 1997). One of the most salient extensions of these techniques is the use of constraints, to focus on the patterns of interest, together with the active use of these constraints to reduce the search space (e.g., Srikant and Agrawal, 1996; Zaki, 2000; Garofalakis et al., 1999; Lee and De Raedt, 2004), and to improve the efficiency of the extractions in practice. Pinto et al. (2001) and Stefanowski and Ziembinski (2005) try to contextualise sequential patterns. However, LSR patterns are different from these context-based sequential patterns. Indeed, Pinto et al. (2001) and Stefanowski and Ziembinski (2005) aim at using a set of attributes to characterise a sequential pattern. The attributes that contextualise sequential patterns do not appear within the sequential patterns whereas neighbourhoods and sequences of LSR patterns are described with the same set of attributes.

6 Conclusions

In this paper, we introduced a new type of pattern for sequential data mining, the LSR pattern. It benefits from synergic action of sequential pattern and rule mining as well as frequent itemset mining. It aims to characterise a sequential pattern by its surrounding context. The order constraint is relaxed in proximity of the sequential pattern in order to discover the frequent itemsets that model its neighbourhood within data sequences. Furthermore, we have shown the relevance of LSR by considering the biomedical NER problem in which sequential patterns and sequential rules present limitations. LSR patterns offer a good trade-off between the high recall of sequential patterns and the high precision of sequential rules for this problem. Indeed, LSR patterns provide a surrounding context awareness that enables the disambiguation of the sequence, thanks to the analysis of its neighbourhood. Experiments, carried out on real datasets, show in these non-trivial cases, the power of our approach. Note that the use of LSR patterns for NER problems leads to an entirely automatic method in which extraction rules can be highly understood by a non-expert. Moreover, LSR patterns can be employed in other domains for the NER problem without effort since the method only considers sequences of tokens on its input.

There are several directions that can be followed to extend the ideas reported in this paper. Concerning the use of LSR in the NER problem, it would be interesting to consider richer input data. Instead of only considering sequences of tokens, we can introduce pieces of information as stemmas or part-of-speech analysis from computational linguistic. We are convinced that considering such pieces of information would result in an additional gain in both recall and precision when applying LSR patterns to the NER problem. It would also be interesting to use some LSR patterns as features in based-ML methods.

We argue that LSR patterns can also be used in many other contexts and problems. As an example, another use of LSR pattern could be the analysis of network datagrams in

the field of network security: an attack could be represented by (l, s, r) where l and r are the surrounding contexts before and after the attack. In this case, l is obviously more important than r in order to prevent the attack. It would also be interesting to apply LSR pattern mining to the discovery of interactions between genes and proteins so as to combine such knowledge with the one discovered in other types of data such as micro array datasets.

Acknowledgements

This work is partly supported by the ANR (French National Research Agency) funded project Bingo2 ANR-07-MDCO-014 (http://www.bingo2.greyc.fr). The work of Jiří Kléma was supported by Czech Ministry of Education under the Programme MSM 6840770012 Transdisciplinary Biomedical Engineering Research II. The Czech-French PHC Barrande project 'Heterogeneous data fusion for genomic and proteomic knowledge discovery' financed the travel expenses.

References

Agrawal, R. and Srikant, R. (1995) 'Mining sequential patterns', *Proc. of the 11th Int. Conf. on Data Engineering (ICDE'95)*, pp.3–14.

Cakmak, A. and Özsoyoglu, G. (2007) 'Annotating genes using textual patterns', in R.B. Altman, A.K. Dunker, L. Hunter, T. Murray and T.E. Klein (Eds.): *Pacific Symposium on Biocomputing*, World Scientific, pp.221–232.

Califf, M.E. and Mooney, R.J. (1999) 'Relational learning of pattern-match rules for information extraction', *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, AAAI, pp.328–334.

Chang, C-H., Kayed, M., Ramzy, M. and Shaalan, K.F. (2006) 'A survey of web information extraction systems', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 10, pp.1411–1428.

Charnois, T., Durand, N. and Kléma, J. (2006) 'Automated information extraction from gene summaries', *Proceedings of the ECML/PKDD Workshop on Data and Text Mining for Integrative Biology*, Berlin, Germany, pp.4–15.

Cohen, A.M. and Hersh, W.R. (2005) 'A survey of current work in biomedical text mining', *Brief Bioinform*, Vol. 6, No. 1, pp.57–71.

Cohen, B.K. and Hunter, L. (2004) 'Natural language processing and systems biology', *Artificial Intelligence Methods and Tools for Systems Biology*, Springer, pp.147–173.

De Raedt, L., Jager, M., Lee, S.D. and Mannila, H. (2002) 'A theory of inductive query answering', *Proceedings of the IEEE Conference on Data Mining (ICDM'02*), Maebashi, Japan, pp.123–130.

Dingare, S., Finkel, J., Nissim, M., Manning, C. and Alex, B. (2004) 'Exploring the boundaries: gene and protein identification in biomedical text', *Proceedings of the BioCreative (Critical Assessment of Information Extraction Systems in Biology) Workshop 2004*, Granada, Spain.

Fukuda, K., Tamura, A., Tsunoda, T. and Takagi, T. (1998) 'Toward information extraction: identifying protein names from biological papers', *Pacific Symposium Biocomputing (PSB'98)*, Hawaii, pp.362–373.

Garofalakis, M., Rastogi, R. and Shim, K. (1999) 'Spirit: sequential pattern mining with regular expression constraints', *Proc. of the 25th Int. Conf. on Very Large Databases (VLD'99)*, pp.223–234.

Humphreys, K., Demetriou, G. and Gaizauskas, R. (2000) 'Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures', *Pacific Symposium Biocomputing*, Hawaii, pp.505–516.

Kim, J-H., Mithell, A., Attwood, T.K. and Hilario, M. (2007) 'Learning to extract relations for protein annotation', *BioInformatics*, Vol. 23, pp.253–257.

Kinoshita, S., Cohen, K.B., Ogren, P.V. and Hunter, L. (2005) 'Biocreative task 1a: entity identification with a stochastic tagger', *BMC Bioinformatics*, Vol. 6, Supp. 1.

Lee, D.S. and De Raedt, L. (2004) 'An efficient algorithm for mining string databases under constraints', *Knowledge Discovery in Inductive Databases 3rd Int. Workshop KDID'04*, Revised selected and invited papers, Springer-Verlag LNCS 3377, pp.108–129.

Leleu, M., Rigotti, C., Boulicaut, J-F. and Euvrard, G. (2003) 'Constraint-based mining of sequential patterns over datasets with consecutive repetitions', in N. Lavrac, D. Gamberger, H. Blockeel and L. Todorovski (Eds.): *PKDD*, Vol. 2838 of Lecture notes in Computer Science, pp.303–314, Springer.

Leser, U. and Hakenberg, J. (2005) 'What makes a gene name? Named entity recognition in the biomedical literature', *Briefings in Bioinformatics*, Vol. 6, No. 4, pp.357–369.

Mannila, H., Toivonen, H. and Verkamo, A. (1997) 'Discovery of frequent episodes in event sequences', *Data Mining and Knowledge Discovery*, Vol. 1, No. 3, pp.259–298.

Meger, N. and Rigotti, C. (2004) 'Constraint-based mining of episode rules and optimal window sizes', *Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Springer-Verlag, LNAI 3202, pp.313–324.

Nanni, M. and Rigotti, C. (2007) 'Extracting trees of quantitative serial episodes', *Knowledge Discovery in Inductive Databases 5th Int. Workshop KDID''06*, Revised selected and invited papers, Springer-Verlag, LNCS 4747, pp.170–188.

Nédellec, C., Bessieres, P., Bossy, R., Kotoujansky, A. and Manine, A-P. (2006) 'Annotation guidelines for machine learning-based named entity recognition in microbiology', *Proceedings of the Data and Text Mining in Integrative Biology Workshop, ECML/PKDD*, Berlin, pp.40–54.

Ng, R.T., Lakshmanan, V.S., Han, J. and Pang, A. (1998) 'Exploratory mining and pruning optimizations of constrained associations rules', *Proceedings of ACM SIGMOD'98*, ACM Press, pp.13–24.

Pei, J., Han, B., Mortazavi-Asl, B. and Pinto, H. (2001) 'Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth', *Proc. of the 17th Int. Conf. on Data Engineering (ICDE'01)*, pp.215–224.

Pei, J., Han, J. and Wang, W. (2002) 'Mining sequential patterns with constraints in large databases', *CIKM*, ACM, pp.18–25.

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q. and Dayal, U. (2001) 'Multi-dimensional sequential pattern mining', *CIKM*, ACM, pp.81–88.

Schmid, H. (1994) 'Probabilistic part-of-speech tagging using decision trees', *International. Conference on New Methods in Language Processing*.

Smith, L., Tanabe, L., Ando, R., Kuo, C., Chung, I., Hsu, C., Lin, Y., Klinger, R., Friedrich, C., Ganchev, K., Torii, M., Liu, H., Haddow, B., Struble, C., Povinelli, R., Vlachos, A., Baumgartner, W., Hunter, L., Carpenter, B., Tsai, R., Dai, H., Liu, F., Chen, Y., Sun, C., Katrenko, S., Adriaans, P., Blaschke, C., Torres, R., Neves, M., Nakov, P., Divoli, A., Maña López, M., Mata, J. and Wilbur, W. (2008) 'Overview of biocreative ii gene mention recognition', *Genome Biology*, Suppl. 2, Vol. S2, No. 9.

Soulet, A. and Crémilleux, B. (2005) 'An efficient framework for mining flexible constraints', in H.T. Bao, D. Cheung and H. Liu (Eds.): *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, Hanoi, Vietnam, Springer, Vol. 3518 of LNAI, pp.661–671.

Srikant, R. and Agrawal, R. (1996) 'Mining sequential patterns: generalizations and performance improvements', in P.M.G. Apers, M. Bouzeghoub and G. Gardarin (Eds.), *EDBT*, Vol. 1057 of Lecture notes in Computer Science, pp.3–17, Springer.

Stefanowski, J. and Ziembinski, R. (2005) 'Mining context based sequential patterns', in P.S. Szczepaniak, J. Kacprzyk and A. Niewiadomski (Eds.): *AWIC*, Vol. 3528 of Lecture notes in Computer Science, pp.401–407, Springer.

Tanabe, L., Xie, N., Thom, L., Matten, W. and Wilbur, J. (2005) 'GENETAG: a tagged corpus for gene/protein named entity recognition', *BMC Bioinformatics*, Vol. 6, p.10.

Ukkonen, E. (1995) 'On-line construction of suffix trees', *Algorithmica*, Vol. 14, No. 3, pp.249–260.

Van Rijsbergen, C.J. (1979) *Information Retrieval*, 2nd ed., Dept. of Computer Science, University of Glasgow.

Yeh, A., Morgan, A., Colosimo, M. and Hirschman, L. (2005) 'BioCreAtIvE task 1A: gene mention finding evaluation', *BMC Bioinformatics*, Vol. 6, p.10.

Zaki, M. (2000) 'Sequence mining in categorical domains: incorporating constraints', *Proc. of the 9th Int. Conf. on Information and Knowledge Management (CIKM'00)*, pp.422–429.