Sequential Patterns to Discover and Characterise Biological Relations

Peggy Cellier, Thierry Charnois, Marc Plantevit

Sequential Patterns to Discover and Characterise Biological Relations

Peggy Cellier[1], Thierry Charnois[1], and Marc Plantevit[2]

[1] Université de Caen, CNRS
Université de Caen, GREYC, UMR6072, F-14032, France
firstname.lastname@info.unicaen.fr
[2] Université de Lyon, CNRS
Université de Lyon 1, LIRIS, UMR5205, F-69622, France
marc.plantevit@liris.cnrs.fr

Abstract. In this paper, we present a method to automatically detect and characterise interactions between genes in biomedical literature. Our approach is based on a combination of data mining techniques: frequent sequential patterns filtered by linguistic constraints and recursive mining. Unlike most Natural Language Processing (NLP) approaches, our approach does not use syntactic parsing to learn and apply linguistic rules. It does not require any resource except the training corpus to learn patterns.

The process is in two steps. First, frequent sequential patterns are extracted from the training corpus. Second, after validation of those patterns, they are applied on the application corpus to detect and characterise new interactions. An advantage of our method is that interactions can be enhanced with modalities and biological information.

We use two corpora containing only sentences with gene interactions as training corpus. Another corpus from PubMed abstracts is used as application corpus. We conduct an evaluation that shows that the precision of our approach is good and the recall correct for both targets: interaction detection and interaction characterisation.

1 Introduction

Literature on biology and medicine represents a huge amount of knowledge: more than 19 million publications are currently listed in PubMed repository[1]. A critical challenge is then to extract relevant and useful knowledge dispersed in such collections. Natural Language Processing (NLP), in particular Information Extraction (IE), and Machine Learning (ML) approaches have been widely applied to extract specific knowledge, for example biological relations. The need for linguistic resources (grammars or linguistic rules) is a common feature of the IE methods. That kind of approach applies rules such as regular expressions for surface searching [5] or syntactic patterns [14, 4]. However rules are handcrafted, those methods are thus time consuming and very often devoted to a

[1] http://www.ncbi.nlm.nih.gov/pubmed/

specific corpus. In contrast, machine learning based methods, for example support vector machines or conditional random fields [8], are less time consuming than NLP methods. They give good results, but they need many features and their outcomes are not really understandable by a user and not usable in NLP systems as linguistic patterns. A good trade-off is the cross-fertilization of IE and ML techniques which aims at automatically learning the linguistic rules [10, 17]. However in most cases the learning process is done from the syntactic parsing of the text. Therefore, the quality of the learned rules relies on syntactic process results. Some works such as [6] do not use syntactic parsing and learn surface patterns using sequence alignment of sentences to derive "motifs". That method allows only interaction patterns to be learned and no new terms to be discovered. Indeed, it is based on a list of terms that represent interactions. In contrast, our proposed approach automatically discovers patterns of interactions and their characterisations (e.g., kind of interaction, modality). In particular, terms representing interactions (and characterisations) are automatically extracted from texts without other knowledge. From our best knowledge, there is no method that in the same time extract interactions and their characterisations.

In this paper, we aim at showing the benefit of using data mining methods [1] for Biological Natural Language Processing (BioNLP). Data mining allows implicit, previously unknown, and potentially useful information to be extracted from data [3]. We present an approach based on frequent sequential patterns [1], a well-known data mining technique, to automatically discover linguistic rules. The sequential pattern is a paradigm more powerful than n-grams. Indeed, n-gram can be seen as a specific instance of sequential pattern. A drawback of n-grams is that the size of the extracted patterns is set for all patterns to n whereas in sequential pattern mining, discovered patterns can have different sizes. In addition, items (i.e. words of texts) within sequential patterns are not necessarily contiguous. Unlike most NLP approaches, the proposed method does not require syntactic parsing to learn linguistic rules and to apply them. In addition, no resources are needed except the training corpus. Moreover, rules coming from sequential patterns are understandable and manageable by an expert.

The process proposed in this paper is in two steps. First, frequent sequential patterns are automatically extracted from the training corpus. In addition, constraints and recursive mining [2] are used to give prominence to the most significant patterns and to filter the specific ones. The goal is to retain frequent sequential patterns which convey linguistic regularities (e.g., entity named relations). Second, after a selection and categorisation of those patterns by an expert, they are applied on the application corpus. The approach is used for the detection of new gene interactions in biomedical literature. An advantage of our method is that interactions can be enhanced with modalities and biological information. Note that no knowledge except the training corpus is used. In addition, in the training corpus the interactions are not annotated.

The paper is organized as follow. Section 2 presents the approach to compute the linguistic rules that allow gene interactions to be extracted and characterised. Section 3 gives an evaluation of our method on biomedical papers from PubMed.

2 Sequential Patterns for Information Extraction

In this section, we introduce sequential patterns defined by Agrawal *et al.* [1]. We explain how we use sequential patterns to extract potential linguistic extraction rules to discover interactions and to identify modalities and biological situations. Linguistic constraints and recursive mining are then presented to reduce the number of extracted patterns. Finally, we show the selection and categorisation of extracted sequential patterns.

2.1 Sequential Patterns

Sequential pattern mining is a well-known technique introduced by Agrawal *et al.* [1] that finds regularities in sequence data. There exist a lot of algorithms that efficiently compute frequent sequential patterns [18, 13, 20].

A *sequence* is an ordered list of literals called *items*, denoted by $\langle i_1 \dots i_m \rangle$ where $i_1 \dots i_m$ are items. A sequence $S_1 = \langle i_1 \dots i_n \rangle$ is *included* in a sequence $S_2 = \langle i'_1 \dots i'_m \rangle$ if there exist integers $1 \leq j_1 < ... < j_n \leq m$ such that $i_1 = i'_{j_1}$, ..., $i_n = i'_{j_n}$. S_1 is called a *subsequence* of S_2. S_2 is called a *supersequence* of S_1. It is denoted by $S_1 \preceq S_2$. For example the sequence $\langle a\ b\ c\ d \rangle$ is a supersequence of $\langle b\ d \rangle$: $\langle b\ d \rangle \preceq \langle a\ b\ c\ d \rangle$.

Table 1. SDB_1, a sequence database

Sequence ID	Sequence
1	$\langle a\ b\ c\ d \rangle$
2	$\langle b\ d\ e \rangle$
3	$\langle a\ c\ d\ e \rangle$
4	$\langle a\ d\ c\ b \rangle$

A sequence database SDB is a set of tuples (sid, S), where sid is a sequence identifier and S a sequence. Table 1 gives an example of database, SDB_1, that contains four sequences. A tuple (sid, S) *contains* a sequence S_α, if S_α is a subsequence of S. The *support*[2] of a sequence S_α in a sequence database SDB is the number of tuples in the database containing S_α: $sup(S_\alpha) = |\{(sid, S) \in SDB \mid (S_\alpha \preceq S)\}|$ where $|A|$ represents the cardinality of A. For example, in SDB_1 $sup(\langle b\ d \rangle) = 2$. Indeed, sequences 1 and 2 contain $\langle b\ d \rangle$. A frequent *sequential pattern* is a sequence such that its support is greater or equal to the support threshold: *minsup*. The sequential pattern mining extracts all those regularities which appear in the sequence database.

2.2 Extraction of Sequential Patterns in Texts

For the extraction of sequential patterns from biological texts, we use a training corpus which is a set of sentences that contain interactions and where the

[2] Sometimes the relative support is used:
$sup(S_\alpha) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \wedge (S_\alpha \preceq S)\}|}{|SDB|}$.

Table 2. Excerpt of the sequence database

Sequence ID	Sequence
...	...
S1	⟨ *here@rb we@pp show@vvp that@in/that AGENE@np ,@, in@in synergy@nn with@in AGENE@np ,@, strongly@rb activate@vvz AGENE@np expression@nn in@in transfection@nn assay@nns .@sent* ⟩
S2	⟨ *the@dt AGENE@np -@: AGENE@np interaction@nn be@vbd confirm@vvn in@in vitro@nn and@cc in@in vivo@rb .@sent* ⟩
...	...

genes are identified. In this paper we consider sentences containing interactions and at least two gene names to avoid problems introduced by the anaphoric structures [21].

From those sentences, sequential patterns representing gene interactions are extracted. The items are the combination of the lemma and their POS tag. The sequences of the database are the interaction sentences where each word is replaced by the corresponding item. The order relation between items in a sequence is the order of words within the sentence. For example, let us consider two sentences that contain gene interactions:

- " *Here we show that <Gene SOX10>, in synergy with <Gene PAX3>, strongly activates <Gene MITF > expression in transfection assays.*"
- " *The <Gene Menin>-<Gene JunD> interaction was confirmed in vitro and in vivo.*"

The gene names are replaced by a specific item, *AGENE@np*, and the other words are replaced by the combinations of the lemmas and their POS tag. An excerpt of the database that contains the sequences associated to those two sentences is given Table 2.

The choice of the support threshold *minsup* is a well-known problem in data mining. With a high *minsup*, only few very general patterns can be extracted. With a low *minsup*, a lot of patterns can be found. In our application, some interesting words, for example "interaction", are not very frequent so that we set a low value of *minsup*. As a consequence, a huge set of patterns is discovered and it needs to be filtered in order to return only interesting and relevant patterns.

2.3 Constraints and Recursive Mining

We use a combination of data mining methods which are well-known to select the most interesting and promising patterns [12, 2]. The constraint-based pattern paradigm enables one to discover patterns under user-defined constraints in order to drive the mining process towards the user objectives. Recursive mining gives prominence to the most significant patterns and filters the specific ones.

Linguistic Constraints. In data mining, the constraints allow the user to define more precisely what should be considered as interesting. Thus, the most

commonly used constraint is the constraint of frequency (*minsup*). However, it is possible to use different constraints instead of the frequency [11]. We use three constraints on sequential patterns to mine gene interactions.

The first constraint is that the pattern must contain two named entities (C_{2ne}). $SAT(C_{2ne})$ represents the set of patterns that satisfy C_{2ne}:
$SAT(C_{2ne}) = \{S = \langle i_1 i_2 \dots i_m \rangle \mid |\{j \ s.t. \ i_j = AGENE@np\}| \geq 2\}$.

The second constraint is that the pattern must contain a verb or a noun (C_{vn}). $SAT(C_{vn})$ represents the set of patterns that satisfy C_{vn}:
$SAT(C_{vn}) = \{S = \langle i_1 i_2 \dots i_m \rangle \mid \exists \ i_j, \ verb(i_j) \ or \ noun(i_j)\}$ where $verb(i)$ (resp. $noun(i)$) is a predicate that returns true if i is a verb (resp. noun).

The last constraint is that the pattern must be *maximal* ($\mathcal{C}_{max}$). A frequent sequential pattern, S_1, is maximal if there is no other frequent sequential pattern, S_2, such that $S_1 \preceq S_2$. $SAT(\mathcal{C}_{max})$ represents the set of patterns that satisfy C_{max}:
$SAT(\mathcal{C}_{max}) = \{s \mid support(s) \geq minsup \wedge \nexists s' \ s.t. \ support(s') \geq minsup, \ s \preceq s'\}$. That last constraint allows the redundancy between patterns to be reduced.

All contrainsts can be grouped in only one constraint $\mathcal{C}_G$ which is a conjunction of previously presented constraints. $SAT(C_G)$ is the set of patterns satisfying C_G.

Recursive Mining. Even if the new set of sequential patterns, $SAT(\mathcal{C}_G)$, is significantly smaller, it can still be too large to be analysed and validated by a human user. Therefore we use *recursive mining* [2] to give prominence to the most significant patterns and to filter the specific ones.

The key idea of recursive pattern mining [2] is to reduce the output by successively repeating the mining process in order to preserve the most significant patterns. More precisely, for each step, the previous result is considered as the new dataset. That recursive process is ended when the result becomes stable.

We divide $SAT(\mathcal{C}_G)$ into several subsets E_{X_i} where the subset E_{X_i} is the set of all sequential patterns of $SAT(\mathcal{C}_G)$ containing the item X_i. More formally, $E_{X_i} = \{s \in SAT(\mathcal{C}_G) \mid \langle X_i \rangle \preceq s\}$. Note that X_i are elements labeled as a verb or a noun. Indeed, we want to identify at least one pattern by verb or noun that appears in the sequential patterns. All verbs and nouns are thus used.

The most k ($k > 1$) representative elements for each E_{X_i} are then computed. Each subset E_{X_i} is recursively mined with *minsup* equals to $\frac{1}{k}$ in order to extract frequent sequential patterns satisfying $\mathcal{C}_G$ previously introduced. The recursion stops[3] when the number of extracted sequential patterns satisfying $\mathcal{C}_G$ is less than or equal to k. It means that the extracted sequential patterns become the sequences of the new database to mine. This process ends when the number of extracted patterns is less than or equal to k. For each subset E_{X_i}, the k extracted sequential patterns are frequent sequential patterns in the first database with respect to *minsup*.

[3] The constraint $\mathcal{C}_{max}$ ensures ending of recursion.

At the end of that step, the number of sequential patterns is controlled. It is less than or equal to $n \times k$ where n is the number of subsets E_{X_i} in $SAT(\mathcal{C}_G)$. Note that k is set *a priori* by the user. Thus, the number of sequential patterns allows them to be analysed by a human user. The sequential patterns are then validated by the user and considered as linguistic information extraction rules for the detection of interactions between genes and their modalities or biological situation. Moreover, it is interesting to note that the subcategorisation of the verb given by the POS tagging indicates the passive or active verb and identifies the direction of the interaction. Prepositions can also allow that kind of information to be found when the pattern does not contain a verb.

2.4 Selection and Categorisation of Patterns

After the extraction of sequential patterns, a human user analyses them as information extraction rules. Some extracted patterns, which are not relevant for interaction detection or characterisation, are removed. The other patterns are selected as information extraction rules. A selected pattern is classified with respect to the kind on information that can be extracted with that pattern. Figure 1 shows the taxonomy that we define and use in our experiments with biological texts. That taxonomy is defined by observation of the extracted patterns. It can be completed with other classes if other kinds of information extraction rules are found. There are three main classes of patterns.

The first class is *interaction patterns* that allows interactions between genes to be found.

The second class is *modality patterns* that allows modalities of interactions to be found. Modalities induce the confidence in the detected interactions. For example, the sentence "It suggests that <gene_name=MYC> interacts with <gene_name=STAT3>." has a lower confidence than "It was demonstrated that <gene_name=MYC> interacts with <gene_name=STAT3>.". We define four levels of confidence: *Assumption*, *Observation*, *Demonstration* and *Related work*, and another subclass representing the *Negation*. A negation modality pattern is for example "AGENE@np absence AGENE@nn".

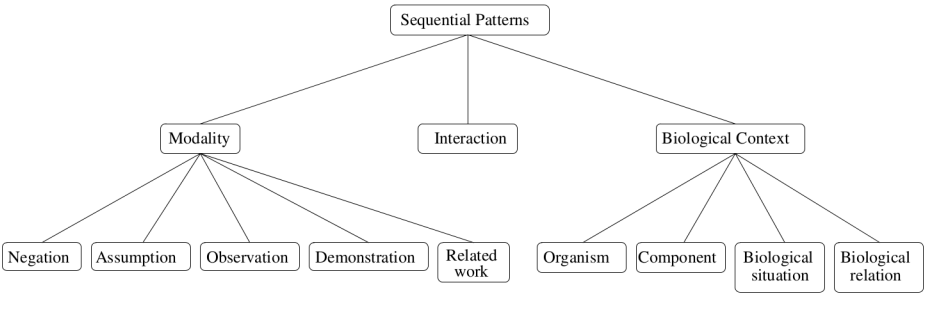

Fig. 1. Taxonomy for pattern selection

The last class is *biological context patterns* that allow information about the biological context of interactions, for example the disease or the organism involved in the interaction, to be found. That class has four subclasses: *organism*, *component*, *biological situation* and *biological relation*. The subclass *organism* enables the organisms involved in the interaction to be found, for example "mouse" or "human". The subclass *component* enables the biological components (e.g. "breast" or "fibroblast") to be detected. The subclass *biological situation* enables to give the framework of interactions, for example, "cancer", "tumor" or "in vitro". The last subclass enables to give the type of biological relation when it is possible, for example "homology".

When the human user has selected and classified all patterns in the different categories, they are applied as extraction rules on the application corpus to discover and characterise new interactions. Note that detection with sequential patterns representing interaction, modalities or biological context is much more elaborated than just a cooccurence detection. Indeed, the order of the words and the context are important, for example $\langle$*these@dt suggest@vvp AGENE@np AGENE@np*$\rangle$ or $\langle$*AGENE@np with@in AGENE@np in@in Vitro@np .@sent*$\rangle$.

3 Experiments

We conducted experiments with our method in order to discover interactions between genes in biological and medical papers. In this section, we present the extraction and validation of linguistic patterns for gene interaction detection and characterisation, and then the application of the selected patterns on a real dataset.

3.1 Extraction Rules

Training Data. Genes can interact with each other through the proteins they synthesize. Moreover, although there are conventions, biologists generally do not distinguish in their papers between the gene name and the protein name synthesized by the gene. Biologists know in context if the sentence is about the protein or gene. Thus, to discover the linguistic patterns of interactions between genes, we merge two different corpus containing genes and proteins.

The first corpus contains sentences from PubMed abstracts, selected by Christine Brun[4] as sentences that contain gene interactions. It contains 1806 sentences. That corpus is available as a secondary source of learning tasks "Protein-Protein Interaction Task (Interaction Award Sub-task, ISS)" from BioCreAtIvE Challenge II [8].

The second corpus contains sentences of interactions between proteins selected by an expert. That dataset, containing 2995 sentences with gene interactions, is described in [15].

[4] Institut de Biologie du Développement de Marseille-Luminy.

Sequential Pattern Extraction. We merged the two datasets previsouly presented and assigned a unique tag for the named entities: *AGENE@np*. A POS tagging is then performed using the *treetagger* tool [16]. The sentences are then ready to extract all the frequent sequential patterns. We set a support threshold, *minsup* equals to 10. It means that a sequential pattern is frequent if it appears in at least 10 sentences (i.e. 0.2% of sentences). Indeed, with that threshold some irrelevant patterns are not taken into account while many patterns of true gene interactions are discovered. Note that other experiments have been conducted with greater *minsup* values (15 and 20). With those greater *minsup* relevant patterns for interaction detection are lost. The number of frequent sequential patterns that are extracted is high. More than 32 million sequences are discovered. Although the number of extracted patterns is high the extraction of all frequent patterns takes only 15 minutes. The extraction tool is *dmt*4 [9].

The application of constraints significantly reduces the number of sequential patterns. Indeed, the number of sequential patterns satisfying the constraints is about $65,000$. However, this number is still prohibitive for analysis and validation by a human expert. Note that, the application of constraints is not time consuming. It takes a couple of minutes.

The recursive mining reduces significantly the number of sequential patterns. The sequential patterns obtained in the previous step are divided into several subsets. The recursive mining of each subset exhibits at most k sequential patterns to represent that subset. In this experiment, we set the parameter k to 4. It allows several patterns to be kept for each noun or verb in order to cover sufficient different cases (for example 4 patterns corresponding to 4 syntactic constructions with the verb *inhibit@vvn* are computed). In the same time it allows the patterns to be analysed by a user. The number of subsets, which are built, is 515 (365 for nouns, 150 for verbs). At the end of the recursive mining, there remains 667 sequential patterns that can represent interactions or their categorisations. That number, which is significantly smaller than previous one, guarantees the feasibility of an analysis of those patterns as information extraction rules by an expert. The recursive mining of those subsets is not time consuming. It takes about 2 minutes.

The 667 remaining sequential patterns were analyzed by two users. They validated 232 sequential patterns for interaction detection and 231 patterns for categorisation of interactions in 90 minutes. It means that 232 sequential patterns represent several forms of interactions between genes. Among those patterns, some explicitly represent interactions. For example, $\langle$*AGENE@np interact@vvz with@in AGENE@np .@sent*$\rangle$, $\langle$*AGENE@np bind@vvz to@to AGENE@np .@sent*$\rangle$, $\langle$*AGENE@np deplete@vvn AGENE@np .@sent*$\rangle$ and $\langle$*activation@nn of@in AGENE@np by@in AGENE@np .@sent*$\rangle$ describe well-known interactions (binding, inhibition, activation). Note that when the patterns are applied, 0 or several words may appear between two consecutive items of the pattern. For example, the pattern $\langle$*AGENE@np interact@vvz with@in AGENE@np .@sent*$\rangle$ matches the sentence "<gene_name=MYC> interacts with <gene_name=STAT3>." and also the sentence "<gene_name=MYC>

interacts with genes in particular <gene_name=STAT3>." Other patterns represent more general interactions between genes, meaning that a gene plays a role in the activity of another gene like $\langle AGENE@np\ involve@vvn\ in@in\ AGENE@np\ .@sent\rangle$, $\langle AGENE@np\ play@vvz\ role@nn\ in@in\ the@dt\ AGENE@np\ .@sent\rangle$ and $\langle AGENE@np\ play@vvz\ role@nn\ in@in\ of@in\ AGENE@np\ .@sent\rangle$. Note that the *"involve"* verb and the *"play role in"* phrase do not belong to the word lists of [19] and [7], also used by Hakenberg *et al.* [6] as terms representing interactions.

The remaining patterns represent modalities or biological context as described in Section 2.4.

The sequential patterns obtained are linguistic rules that can be used on biomedical texts to detect and characterise interactions between genes. Note that to be applied, those patterns do not need a syntactic analysis of the sentence. The process just tries to instantiate each element of the pattern in the sentence.

3.2 Application: Detection and Characterisation of Gene Interactions

In order to test the quality of the sequential patterns found in the previous section, we consider 442,040 biomedical papers from PubMed. In that dataset, the names of genes or proteins are labeled thanks to[5]. We randomly took 200 sentences and tested whether the linguistic patterns can be applied. For each sentence, we manually measure the performance of linguistic sequential patterns to detect those interactions and their characteristics. Note that we also carried out a POS tagging of those sentences in order to correctly apply the pattern language, most of applications of the linguistic sequential patterns is almost instantaneous.

Table 3. Detection and characterisation of interactions

	Precision	Recall	*f-score*
Interaction detection	0.83	0.75	0.79
Interaction categorisation	0.88	0.69	0.77

Table 3 presents the scores of the application of the patterns as extraction rules: Precision, Recall and *f-score*[6]. For the gene interaction detection, the precision is good and the recall is correct. Those results are comparable to the results of other methods in literature, however, we can note that the tasks are not the same [8]. For the interaction characterisation, the precision is good and the recall is about 69%. There are several reasons that explain why the recall is not greater. They are discussed in the next section.

3.3 Discussion

About Interaction Detection. Although the results of the POS tagger tool are mainly correct, there still be some labeling errors on lemmatization or

[5] http://bingotexte.greyc.fr/

[6] The used *f-score* function is : $f\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$.

assignment of a grammatical category. Our method is robust with respect to that phenomenon, indeed those errors are also present in the extracted patterns. Thus, if an error is frequent, it appears in a pattern. For example, *treetagger* does not lemmatize the word *cotransfected* but some extracted patterns contain *cotransfected@vvn*.

Note that for the experiments the scope of extracted linguistic patterns is the whole sentence. That scope may introduce ambiguities in the detection of interactions when more than two genes appear in sentences. Several cases are possible: when several binary interactions are present in the sentence, when the interaction is n-ary ($n \geq 3$) or when an interaction is found with a list of genes. The case of n-ary interactions can be solved with a training dataset containing n-ary interactions. The other two cases can be treated by introducing limitations of pattern scope, for example cue-phrases (e.g. but, however).

False negatives depend on the absence of some nouns or verbs of interaction in the patterns. For example, the noun "modulation" is not learned in a pattern whereas the verb "modulate" appears in patterns. This suggests that the use of linguistic resources (e.g. lexicon or dictionary), manually or semi-automatically, can improve patterns and thus interaction detection.

About Interaction Characterisation. The false negatives, which are dependent on the absence of some patterns, are also an important problem for interaction characterisation. For example, in our experiments in the sentence *"<gene_name=SP1> binding is enhanced by association with <gene_name=CDK2> and <gene_name=CDK2>, both in vivo and in vitro ."* the biological situation "in vitro" is detected whereas "in vivo" is not detected. Indeed, there is no sequential pattern extracted from the training corpus that contains "in vivo". That case is considered as a false negative. The recall (69%) is strongly dependent on the number of false negatives. Note that the false negatives mainly come from missing biological context (about 92%). It is explained by the difficulty to have a training corpus that contains all biological context (e.g. body parts ("liver", "pituitary gland", ...), disaeses). The false negatives due to missing modalities are seldom (about 8%). Those false negatives are explained by the fact that patterns containing "perform" have not been validated by the human users as IE rules whereas those patterns may find some modalities.

4 Conclusion

The proposed approach aims at automatically discovering linguistic IE rules using sequential patterns filtered by linguistic constraints and recursive mining. Unlike existing methods, our approach is independent of syntactic parsing and does not require any resource except the training corpus to learn patterns. Note that in this training corpus interactions are not annotated. In addition, the implementation is simple. The sequential patterns, which are automatically generated, are used as linguistic rules. An advantage of the use of sequential patterns is that they are understandable and manageable IE rules. The expert can easily

modify the proposed rules or add other ones. We illustrated the method on the problem of the detection and characterisation, with some modalities and biological information, of gene interactions. However, the proposed approach can be straightforwardly applied to other domains without additional effort to develop custom features or handcrafted rules.

The experiments related on PubMed annotated corpus show that results are close to other approaches in literature. We are convinced that those results can be easily improved. Indeed, we used directly the discovered patterns as IE rules, without modifying them. Adding or enhancing patterns with expert knowledge, or using a specialized dictionary to enhance manually or semi-automatically the discovered patterns should reduce false negatives (and false positives also). Using heuristics to limit the scope of applied patterns (e.g. cue-phrases) should also improve the precision.

Acknowledgments. The authors would like to thank Christophe Rigotti (Université de Lyon - LIRIS, Fr) for invaluable discussions and for *dmt4*. This work is partly supported by the ANR (French National Research Agency) funded project Bingo2 ANR-07-MDCO-014.

References

[1] Agrawal, R., Srikant, R.: Mining sequential patterns. In: International Conference on Data Engineering (1995)
[2] Crémilleux, B., Soulet, A., Kléma, J., Hébert, C., Gandrillon, O.: Discovering Knowledge from Local Patterns in SAGE data. IGI Publishing (2008)
[3] Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. In: Knowledge discovery in databases, pp. 1–30. AAAI/MIT Press (1991)
[4] Fundel, K., Küffner, R., Zimmer, R.: RelEx - relation extraction using dependency parse trees. Bioinformatics 23(3), 365–371 (2007)
[5] Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference (EACL). The Association for Computer Linguistics (2006)
[6] Hakenberg, J., Plake, C., Royer, L., Strobelt, H., Leser, U., Schroeder, M.: Gene mention normalization and interaction extraction with context models and sentence motifs. Genome biology 9(Suppl. 2) (2008)
[7] Hao, Y., Zhu, X., Huang, M., Li, M.: Discovering patterns to extract protein-protein interactions from the literature: Part ii. Bioinformatics (2005)
[8] Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A.: Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology (2008)
[9] Nanni, M., Rigotti, C.: Extracting trees of quantitative serial episodes. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 170–188. Springer, Heidelberg (2007)

[10] Nédellec, C.: Machine learning for information extraction in genomics - state of the art and perspectives. In: Text Mining and its Applications: Results of the NEMIS Launch Conf. Series: Studies in Fuzziness and Soft Comp. Sirmakessis, Spiros (2004)
[11] Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained association rules. In: SIGMOD Conference (1998)
[12] Pei, J., Han, B., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: Proc. of the 17th Int. Conf. on Data Engineering, ICDE 2001 (2001)
[13] Pei, J., Han, B., Mortazavi-Asl, B., Pinto, H.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. of the 17th Int. Conf. on Data Engineering, ICDE 2001 (2001)
[14] Rinaldi, F., Schneider, G., Kaljurand, K., Hess, M., Romacker, M.: An environment for relation mining over richly annotated corpora: the case of genia. BMC Bioinformatics 7(S-3) (2006)
[15] Rosario, B., Hearst, M.A.: Multi-way relation classification: application to protein-protein interactions. In: Proc. of the conf. on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2005)
[16] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing (September 1994)
[17] Schneider, G., Kaljurand, K., Rinaldi, F.: Detecting protein-protein interactions in biomedical texts using a parser and linguistic resources. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 406–417. Springer, Heidelberg (2009)
[18] Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057. Springer, Heidelberg (1996)
[19] Temkin, J.M., Gilder, M.R.: Extraction of protein interaction information from unstructured text using a context-free grammar. Bioinformatics (2003)
[20] Zaki, M.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning 42(1/2) (2001)
[21] Zweigenbaum, P., Demner-Fushman, D., Yu, H., Cohen, K.B.: Frontiers of biomedical text mining: current progress. Brief Bioinform. (2007)