BIOINFORMATICS

Vol. 23 ISMB/ECCB 2007, pages i256–i263
doi:10.1093/bioinformatics/btm168

Learning to extract relations for protein annotation

Jee-Hyub Kim[1,*], Alex Mitchell[2,3], Teresa K. Attwood[2,3] and Melanie Hilario[1]

[1]Artificial Intelligence Laboratory, University of Geneva, CH-1211 Geneva 4, Switzerland, [2]Faculty of Life Sciences and School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PT and [3]European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

ABSTRACT

Motivation: Protein annotation is a task that describes protein X in terms of topic Y. Usually, this is constructed using information from the biomedical literature. Until now, most of literature-based protein annotation work has been done manually by human annotators. However, as the number of biomedical papers grows ever more rapidly, manual annotation becomes more difficult, and there is increasing need to automate the process. Recently, information extraction (IE) has been used to address this problem. Typically, IE requires pre-defined relations and hand-crafted IE rules or annotated corpora, and these requirements are difficult to satisfy in real-world scenarios such as in the biomedical domain. In this article, we describe an IE system that requires only sentences labelled according to their relevance or not to a given topic by domain experts.

Results: We applied our system to meet the annotation needs of a well-known protein family database; the results show that our IE system can annotate proteins with a set of extracted relations by learning relations and IE rules for disease, function and structure from only relevant and irrelevant sentences.

Contact: jee.kim@cui.unige.ch

1 INTRODUCTION

Proteins are important entities in the biomedical domain; their annotation is a task that describes protein X in terms of topic Y (e.g. disease, function, structure, etc.). Usually, this type of information is gleaned from the biomedical literature.[1] Many protein sequence databases (e.g. Swiss-Prot, PIR-PSD) and protein family databases (PROSITE, PRINTS, InterPro, etc.) have been constructed; protein sequences and families in these databases have been annotated mainly by human curators via a labour-intensive and time-consuming manual processes. However, as the number of published biomedical papers grows ever more rapidly, there is an increasing need to automate protein annotation. Recent attempts to do so vary depending on the granularity of annotation required: document-level (e.g. the TREC genomics tracks), sentence-level (Mitchell *et al.*, 2005; Nedellec *et al.*, 2001), keyword-level (Deng *et al.*, 2004) and relation-level annotation. The first simply annotates proteins with references to relevant documents; the second and third provide sentences and keywords extracted from these documents. Relation-level annotation provides structured information in the form of relations distilled from these documents. Information extraction (IE) has been used to extract relations, and IE systems for relation-level protein annotation have been developed for different topics, such as protein structure, protein–protein interaction and phosphorylation (Gaizauskas *et al.*, 2003, Hu *et al.*, 2005; Saric *et al.* 2006). Nevertheless, many topics remain untapped: e.g. cell cycle, tissue specificity.

In general, the development of IE systems consists of two main steps: pre-defining relations and developing IE rules; both steps are difficult for biologists, who tend to have little or no formal IE training. With regard to pre-defining relations, it is hard to specify precisely all possible relations to extract, especially in complex and dynamically evolving domains such as the biomedical domain. Once relations are defined, IE rules for extracting these relations are developed using either knowledge engineering (KE) or machine learning (ML) approaches. In the KE-based approach, rules are written manually by knowledge engineers with the help of domain experts (i.e. biologists). However, this approach is not scalable and incurs high maintenance costs. On the other hand, the ML-based approach typically has relied on annotated corpora to learn IE rules for defined relations. Building annotated corpora is also a daunting task for biologists. All these difficulties have prevented biologists from building and using IE systems for new topics in which they are interested.

Compared with typical IE system development methods mentioned above, our method requires only sentences labelled by biologists as relevant or not to their selected topics; it requires neither pre-defined relations, nor knowledge engineers nor annotated corpora. Simply collecting sentences is a much easier task than meeting the requirements of the previous methods. In this article, we describe an ML-based IE system development method that helps biologists define relations of interest and extracts these defined relations from text, given only sentences provided by biologists.

2 RELATED WORK

Previous work can be assigned to several categories, depending on the availability of pre-defined relations and how IE rules are developed. We start from work done with pre-defined relations, ending with work done without them.

Given pre-defined relations, the only thing to be done is to develop IE rules. For this, there are four different methods: writing IE rules manually, learning IE rules from

*To whom correspondence should be addressed.

[1] Protein annotation also has been done by sequence-based methods, but in this paper we only focus on literature-based methods.

annotated corpora, learning IE rules from pre-labelled corpora,[2] and learning IE rules from raw corpora (i.e. neither annotated nor pre-labelled). In the past decade, a significant amount of work has been done on *learning from annotated corpora* (Califf and Mooney, 2003; Freitag, 2000; Soderland, 1999). As annotating corpora is a difficult and time-consuming job, methods for *learning from pre-labelled corpora* and *learning from raw corpora* have been tested. In the former, a set of IE rules have been learned from relevant and irrelevant text examples, and these learned rules have been mapped to pre-defined relations by domain experts (Riloff, 1996). In the latter, pairs of named entities have been clustered with contextual words into pre-defined relations (Hasegawa *et al.*, 2004).

We move on to other categories where pre-defined relations—and hence pre-annotated corpora—are not provided by the user. Here, two methods can be considered: *learning relations and IE rules from raw corpora*, and *learning relations and IE rules from pre-labelled corpora*. The first method, which tackles the most difficult setting, has been introduced by Collier without providing a proof-of-concept (Collier, 1996). The method presented in this article belongs to the second category, where the difficulty is reduced from using raw corpora to using pre-labelled corpora.

3 PROBLEM AND APPROACH

The goal of the work reported in this article is to alleviate the burden of developing IE systems. To achieve this goal, we use only sentences. Our problem can be formally defined as follows: *given relevant sentences that describe protein X in terms of any topic Y and irrelevant sentences, learn to extract relations for protein annotation*. From this problem definition, it should be noted that there are two sub-problems to be solved: *identifying relations* and *learning IE rules*. For the former we need to know *what to extract* from relevant and irrelevant sentences, whereas for the latter we need to find *how to extract* those interesting relations. To solve these two sub-problems simultaneously, we applied a *bottom-up* approach. This approach can be summarized as follows. First, learn all possible IE rules that discriminate relevant from irrelevant sentences. Second, ask users to select IE rules of interest from these learned IE rules. Third, transform the selected IE rules and group them into relations. As a result, a set of IE rules are mapped onto each relation. Figure 1 shows our approach with some examples.

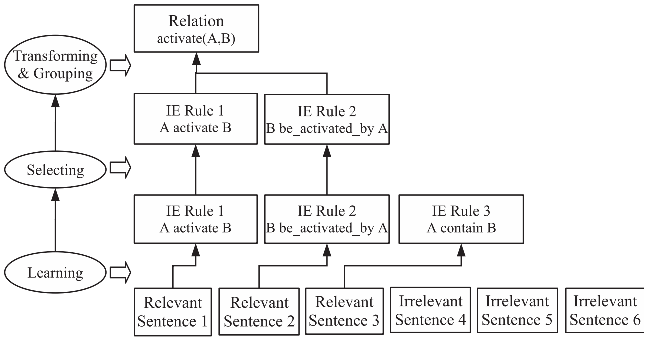


Fig. 1. Bottom-up approach.

The idea behind this approach is that although we do not have any pre-defined relations, there are some linguistic patterns occurring more frequently in relevant sentences than in irrelevant sentences, and those patterns have a high possibility of expressing relations of interest to users for a given topic.

In our approach, users are involved in two stages: providing relevant and irrelevant sentences, and selecting IE rules. We believe this is much easier than pre-specifying relations and writing rules or annotating corpora. It is important that the output of an ML algorithm (i.e. learned IE rules) be readable and interpretable by the domain experts, so that they can select IE rules for each topic. For this reason, we selected inductive logic programming (ILP) (Muggleton and Raedt, 1994) as our ML framework. ILP uses first-order logic (FOL) as a representational language, and FOL is both machine- and human-readable. In ILP a logic program is induced from examples. Developed originally as a programming assistant, ILP was meant to allow the programmer to modify learned rules; this feature fits well with our purpose. ILP has been successfully applied in bioinformatics tasks, e.g. in protein structure prediction (Cootes *et al.*, 2003) and in systems biology (Lodhi and Muggleton, 2004).

The following section describes how our IE system has been developed, guided by the approach mentioned above.

4 METHODS

Our IE system takes as input relevant and irrelevant sentences and gives as output a set of extracted relations. The system consists of four main modules, as shown in Figure 2. Each module is described in detail in the following subsections.

4.1 Analysing sentences

The first process in our IE system is sentence analysis. For this, we applied the Memory-Based Shallow Parser (MBSP) (Daelemans *et al.*, 1999), which has been adapted to the biological domain on the basis of the GENIA corpus (Kim *et al.*, 2003). The performance of the MBSP on the GENIA corpus is as follows: an overall accuracy of 97.6% on POS tagging, and 71.0% on protein named entity tagging.

The MBSP consists of a number of different text analysis modules: tokenization, part-of-speech (POS) tagging, concept tagging (based on the GENIA ontology[3]), chunking, PNP-finding and grammatical function assignment (subject, object, time, location, etc.); each module provides its own type of information. All these types of information are used for learning IE rules, as discussed in the next subsection. Table 1 shows an example sentence analysed by the MBSP.

4.2 Learning IE rules

To learn all possible *candidate* IE rules from analysed relevant and irrelevant sentences, we utilized ILP. In the ILP framework, a hypothesis H (i.e. a set of rules) is induced from examples E with

[2] Pre-labelled corpora are made of texts labelled relevant or irrelevant by users.

[3] We used the following tags: protein, cell-type, DNA-part, cell-line, virus, protein-part, DNA, protein-complex, lipid, multi-cell, tissue, cell-component, RNA, etc.

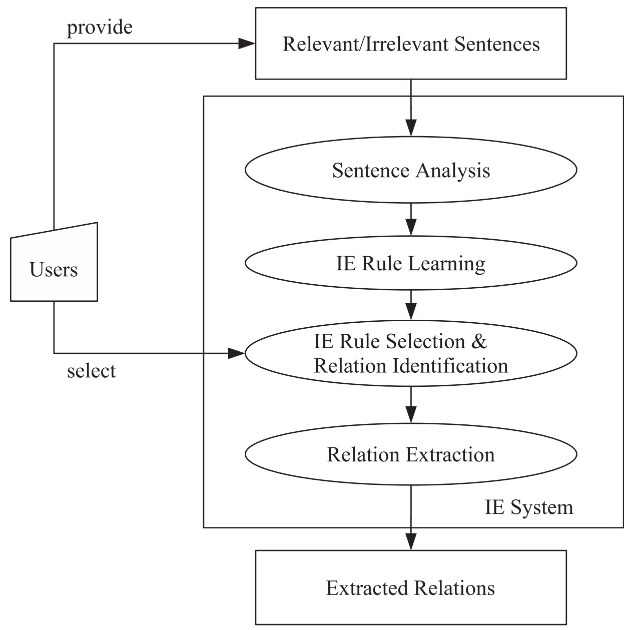


Fig. 2. Overall IE system architecture. The first three modules are used for building an IE system and the last module for applying it. (oval: module, rectangle: data).

Table 1. An example sentence, 'Examples of this are the RNA-binding protein containing the RNA-binding domain (RBD)', analysed by the MBSP

Chunk	Syntactic	Semantic	SVO relation
Examples	Noun phrase		Subject of 'are'
of	Preposition		
this	Noun phrase		
are	Verb phrase		
the RNA-binding protein	Noun phrase	Protein	Subject of 'contain'
containing	Verb phrase		
the RNA-binding domain (RBD)	Noun phrase	Domain	Object of 'contain'
	.		

Each row represents a chunk with syntactic, semantic, and SVO relation information.

background knowledge B such that H implies examples E with background knowledge B. This is formulated as follows:

$$B \wedge H \models E$$

In order to use this framework to learn a set of IE rules, background knowledge B and examples E were prepared and encoded (in Prolog) as follows. For background knowledge B, we used two types: *linguistic heuristics* and *sentence descriptions*. The role of linguistic heuristics is to guide the construction of single-slot IE patterns. For instance, a heuristic <subject> verb (encoded in line 1 in Fig. 3) constructs a single-slot IE pattern that extracts the subject of a verb and fills up a single slot. The linguistic heuristics used are: <subject> verb, verb <direct object>, verb preposition <noun phrase> and noun preposition <noun phrase>. (Hereafter, subject is referred to as subj and direct object as dobj.) By sentence descriptions, we mean the representation of analysed sentences using the predicates defined in Table 2 (refer to lines 3–11 in Fig. 3). For examples E, positive and negative examples have been encoded as in lines 12–13 in Figure 3.

Given the representation of B and E as described above, we used an ILP system, ALEPH (Srinivasan, 2000) to learn H, that is, a set of IE rules. ALEPH learns a set of rules as follows:

1. Select an example to be generalized. If none exists, stop, otherwise proceed to the next step.
2. Construct the most specific clause that entails the example selected, and is within language restrictions provided.
3. Find a clause more general than the bottom clause.
4. Add the clause with the best score to the current theory, and remove all examples made redundant.

To calculate the score of each rule, we used the *WRAcc* (Weighted Relative Accuracy) measure which is defined as follows. Given a rule $R : Head \leftarrow Body$,

$$WRAcc(R) = coverage(R) \times (accuracy(R) - accuracy(Head \leftarrow true))$$

where, the default rule ($Head \leftarrow true$) predicts all instances to satisfy H. This measure prefers a slightly inaccurate but very general rule. The *WRAcc* measure has been proved successful in discovering interesting rules, each of which is thought to represent a subgroup (Lavrac *et al.*, 2004). In our context, these rules (or subgroups) can be viewed as relations to extract. An example of a learned IE rule is shown in line 14 in Figure 3.

All the learning steps described above can be summarized as follows. Examples (i.e. sentences) are represented as a set of single-slot IE patterns constructed from sentence descriptions by the linguistic heuristics, and the goal of learning is to find a subset of single-slot IE patterns that distinguish relevant and irrelevant sentences for each topic. What should be noted here is that, in the absence of annotated corpora, rule learning has been guided by efficacy in classifying relevant and irrelevant sentences rather than by specific target information to be extracted. We can imagine that some learned IE rules are spurious; user feedback is required to filter them out.

4.3 Selecting IE rules and identifying relations

These learned IE rules were given to the domain experts, together with additional information: the number of sentences covered by a rule, the precision of the rule and the sentences covered by the rule (refer to Fig. 4)[4]. Learned IE rules can be viewed as surface forms of relations. Based on the learned IE rules and their corresponding statistics, the experts selected IE rules they found interesting; the selected rules were transformed into relations using the mapping rules in Table 3. For example, the trigger of an IE rule was transformed into a relation name. In accordance with the mapping table, the rule <subj:*>[X] vp:contain & vp:contain <dobj:domain>[Y] was transformed into the relation contain(X,Y) (shown in line 15 in Fig. 3). Note that traditional IE methods do not need this relation identification step, as relations are pre-defined.

Among selected and transformed relations, it is possible that several relations have the same meaning. For example, relation contain(A,B) has the same meaning as relation be_contained(B,in:A). We performed a rudimentary paraphrase analysis to group identical relations into a representative one. Table 4 shows paraphrase patterns we considered in this article. One of those patterns can deal with relative clauses.

Our current IE system can identify and extract binary relations, and postprocessing rules have been written to map learned IE rules onto binary relations. Extension to arbitrary n-ary relations can be considered and this extension requires writing mapping rules. Despite this

[4] Prolog style rules can be difficult for the users to understand, so more human-friendly rules were given to them.

```
% Linguistic Heuristics

% if sentence S has two contiguous chunks C1 and C2
% where C1 is a noun phrase (np) and C2 is a verb phrase (vp) with a trigger T, and C1 is the subject of C2
% then build an extraction pattern that extracts the subject of a verb.
1: ep(subj_vp,S,C1,T) :- has(S,C1,np,_),has(S,C2,vp,T),subj(C1,C2).

% if sentence S has two contiguous chunks C1 and C2
% where C1 is a verb phrase with trigger T and C2 is a noun phrase, and C2 is the direct object of C2
% then build an extraction pattern that extracts the direct object of a verb.
2: ep(vp_dobj,S,C2,T) :- has(S,C1,vp,T),has(S,C2,np,_),dobj(C2,C1). ...

% Sentence Descriptions of an Analysed Sentence

% "Examples of this are the RNA-binding protein containing the RNA-binding domain (RBD)."
3: s(s7).
4: c(c7_0). has(s7,c7_0,np,'example'). sem(c7_0,'example').
5: c(c7_1). has(s7,c7_1,pp,'of'). sem(c7_1,'of').
6: c(c7_2). has(s7,c7_2,np,'this'). sem(c7_2,'this').
7: c(c7_3). has(s7,c7_3,vp,'be'). sem(c7_3,'be').
8: c(c7_4). has(s7,c7_4,np,'protein'). sem(c7_4,'protein').
9: c(c7_5). has(s7,c7_5,vp,'contain'). sem(c7_5,'contain').
10: c(c7_6). has(s7,c7_6,np,'domain'). sem(c7_6,'domain'). ...
11: subj(c7_0,c7_3). next(c7_0,c7_1). next(c7_1,c7_2). next(c7_2,c7_3). ...

% Positive Examples

12: structure(s7). % Sentence s7 is labelled as relevant.

% Negative Examples

13: :- structure(s8). % Sentence s8 is labelled as irrelevant.

% Learned IE Rules (for the "structure" topic)

% extract the subject and the object of verb "contain", where the semantic type of the object is "domain"
14: structure(S) :- ep(subj_vp,S,C1,contain), ep(vp_dobj,S,C2,contain), sem(C2,domain).

% Transformed IE Rules

% fill up relation "contain" with fillers extracted by the IE rule in Line 14
15: contain(C1,C2) :- ep(subj_vp,S,C1,contain), ep(vp_dobj,S,C2,contain), sem(C2,domain).
```

Fig. 3. Excerpts of problem representation. These excerpts are written in Prolog. Comment lines are introduced with %, and program lines are numbered for readability.

Table 2. Predicates used to represent analysed sentences

Predicate	Argument type	Description
s/1	Sentence (S)	Type declaration for a sentence
c/1	Chunk (C)	Type declaration for a chunk
has/4	S, C, SyntacticRole, HeadWord	States the relation between a sentence and a chunk
next/2	C, C	States that a chunk follows another chunk
sem/2	C, Semantics	States that the semantic type or head word of a chunk
subj/2, dobj/2	C, C	States subject and object relations

restriction, writing a small set of mapping rules can still significantly reduce the burden of handcrafting a huge number of IE rules.

4.4 Applying IE rules to extract relations

Finally, we applied selected IE rules to extract relations from sentences. Below is one example of an IE rule (written in human-friendly form) belonging to the relation *translocate*:

- RULE: <subj:protein>[X] vp:translocate & vp:translocate <from:*>[Y]
- TRIGGER: translocate
- RELATION: translocate(X,Y)

In this example, the rule consists of two single-slot IE patterns with each pattern extracting some part of a given sentence. For instance, the single-slot IE pattern <subj:protein>[X] vp:translocate extracts the subject of a verb phrase with 'translocate' as head, where the subject must be a protein. Once the pattern extracts the subject from the sentence, this value is stored in variable X. Similarly, the second IE pattern vp:translocate <from:*>[Y] extracts the source location of the protein, which can be of any semantic type (*) and stores it in variable Y. A rule is only applied to a given sentence if the sentence

```
RULE: <subj:*>[X] vp:contain & vp:contain <dobj:domain>[Y] // * means any semantic type
STATISTICS: 9, 0.9 // number of sentences covered by the rule, precision of the rule
S1: Myocilin is a secreted glycoprotein that forms multimers and contains a leucine zipper and an olfactomedin domain.
S2: The Bcl-B protein contains four Bcl-2 homology (BH) domains (BH1 , BH2 , BH3 , BH4) and a predicted carboxyl-terminal transmembrane (TM) domain.
...

RULE: <subj:protein>[X] vp:be & np:regulator <of:*>[Y]
STATISTICS: 4, 1
S1: The transcriptional coactivators CBP and p300 are critical regulators of metazoan gene expression.
S2: The PP2B protein phosphatase , also known as calcineurin , is a regulator of ion homeostasis in yeast cells.
...

Structure of a single-slot IE pattern: <Syntactic_Type:Semantic_Type>[Slot_for_Extract] Syntactic_Type:Trigger
```

Fig. 4. Sample IE rules with statistics and covered sentences. The first IE rule means *extract the subject and the object of the verb* `contain`, *where the subject could be any semantic type and the object must be a domain.* The second IE rule means *extract the subject of the verb* be *where the subject must be a protein, and extract the following noun phrase after* regulator of.

Table 3. Mapping between IE rules and relations

IE Rules	Relations
Extract	Argument
Trigger	Relation name
Syntactic tag	Argument position

Table 4. Paraphrase patterns

Pattern	Example
A + verb (active form) + B	A activate B
B + be + *ed* -participle + by + A	B be activated by A
Nominal form (with suffix - *tion*) of verb + of + B + by + A	activation of B by A
A + be + nominal form (with suffix - *or*) of verb + of + B	A is an activator of B
A + be ... that + verb (active form) + B	A is ... that activates B

contains the trigger of the rule. A relation defines what the output looks like. The value stored in X by the extraction pattern is used to fill up slot X in this relation.

The following is an example sentence to which the above IE rule can be applied. Once its trigger translocate is matched against the sentence, the rule is applied to extract the subject and the object of the verb. The extracted values are used to fill up the relation and the final output is shown below.

- INPUT: the zinc finger protein ZPR1 translocates from the cytoplasm to the nucleus after treatment of cells with mitogens.
- OUTPUT: translocate('The zinc finger protein ZPR1', from: 'the cytoplasm').

Another more complex example shows how a relative clause is handled. In this case, the IE rule is applied when its two triggers are matched against an input sentence.

- RULE: < subj:protein > [X] vp:be & vp:mediate < dobj:* > [Y]
- TRIGGER: be, mediate
- RELATION: mediate(X,Y)
- INPUT: We propose that Aer is a flavoprotein that mediates positive aerotactic responses in *Escherichia coli*.
- OUTPUT: mediate('Aer', 'positive aerotactic responses')

5 EXPERIMENTAL RESULTS

We applied our IE system to meet the annotation needs of the PRINTS database[5] (Attwood *et al*., 2003). The task involved extracting relations between proteins and any other biological entities, provided they were relevant to three topics: disease, function and structure. The PRINTS database annotators collected sentences relevant to these topics from MEDLINE abstracts, and double-checked that all other sentences in these abstracts were indeed not relevant to any of the three topics. Table 5 shows the statistics of the three corpora.

We divided each corpus into two sets: 80% for training and 20% for testing (training set: 621, 1014 and 927 relevant sentences for disease, function and structure, respectively; test set: 156, 256 and 232). From the training set for each topic, we learned a set of IE rules, and the annotators selected those of most interest. After transforming and grouping the selected IE rules into relations, we applied a set of IE rules for each relation to the test set.

Once relations were extracted from the test set, they were manually evaluated by the PRINTS annotators. In this evaluation, recall was approximated because the annotators only evaluated extracted relations. It was difficult for them to go through all the sentences to find false negative relations that were missed by our IE system. Assuming only one relation could be extracted from each relevant sentence, the recall rate was approximated by dividing the number of correctly extracted relations by the total number of relevant sentences.

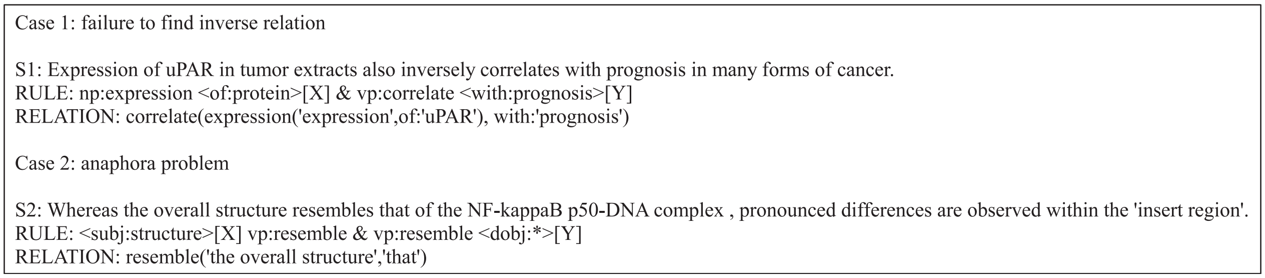
Table 5 shows the evaluation results of our IE system. During the evaluation, we encountered partially correct cases. For instance, Figure 5 shows one example where the extracted relation was the inverse of the actual relation. Other partially correct cases involve problems of anaphora resolution, missing slots, etc. In the computation of both precision and recall, partially correct cases received a score of 0.5 rather than 1. Finally, we examined some false-negative cases and found that

[5] PRINTS is a protein family database.

Table 5. Summary of IE development corpora and evaluation of extracted results

Topic	Corpora			On training set			On test set						
	Positive	Negative	Class distribution	Learned rules	Selected rules	Relations	Relevant sentences	Extracted relations	Correctly extracted relations	Partially correct extracted relation	Precision	Recall	f1-measure
Disease	777	1403	36–64%	55	32	21	156	38	24	9	75	18.3	29.4
Function	1268	2625	33–67%	125	64	23	256	80	38	30	66.3	15.1	24.6
Structure	1159	1750	40–60%	146	76	20	232	166	131	21	85.3	61	71.1

The first three columns (in 'On Training Set') show a snapshot of building our IE system, and the last three columns (in 'On Test Set') shows the performance of the built IE system. The columns between them used to calculate the performance.



Case 1: failure to find inverse relation

S1: Expression of uPAR in tumor extracts also inversely correlates with prognosis in many forms of cancer.
RULE: np:expression <of:protein>[X] & vp:correlate <with:prognosis>[Y]
RELATION: correlate(expression('expression',of:'uPAR'), with:'prognosis')

Case 2: anaphora problem

S2: Whereas the overall structure resembles that of the NF-kappaB p50-DNA complex , pronounced differences are observed within the 'insert region'.
RULE: <subj:structure>[X] vp:resemble & vp:resemble <dobj:*>[Y]
RELATION: resemble('the overall structure','that')

Fig. 5. Partially-correct relations.

Table 6. Interesting relations and entities for each topic

Topic	Relations	Entities
Disease	be_ associated, is_ a, be_ mutated, be_ caused, be_ increased, contribute, be_ deleted	Disorder, marker, disease, cancer
Function	induce, block, mediate, is_ a, belong, act	Role, regulator, inhibitor, family
Structure	contain, form, share, lack, bind, encode, be_ conserved	Domain, motif, site, surface, region

Table 7. Protein annotation results for protein NF-kappaB

Disease	be_ implicated('NF-kappaB', in: 'the pathogenesis')

Function	regulate('IkappaBalpha', 'the transcription factor NF-kappaB')
	activate('BCMA', 'NF-kappaB')

Structure	be_ composed('The transcription factor NF-kappaB', of: 'heterodimeric complexes') form('p65 (RelA)', 'the prototypical NF-kappaB transcription factor complex')

our IE rule representation lacked the expressive power to catch certain relations. For example, it missed relations expressed by 'be + adjective preposition' type (e.g. protein X is important for Y) or by complex noun phrases (e.g. motif Y-containing protein X), which occur quite often in our corpora.

Table 6 shows extracted relations that were judged interesting by the users. We found that the is_a relation was extracted across different topics, although this relation comprised arguments with different semantic information. For example, relation is_a was used as is_a(protein, marker) for disease and as is_a(protein, family) for function. We compared our selected relations for topic *structure* with those defined in the PASTA system (Gaizauskas *et al.*, 2003), and found some new relations such as lack and share. Table 6 shows interesting entities other than proteins used as arguments of selected relations. Finally, Table 7 shows how the same protein can be annotated in terms of different topics.

6 DISCUSSION

Overall, we achieved high precision for all topics, though some topics have low recall. These results show that our IE system can annotate proteins in terms of a given topic by learning relations and IE rules for those relations, based only on relevant and irrelevant sentences. Although it is not possible to compare our method directly with other methods, our work (with an average F1-measure of 41% over 3 topics) may be compared indirectly with recent work on the extraction of regulatory gene/protein networks, where the authors use manually written rules and report an F1-measure of 44% (Saric *et al*., 2006).

The PRINTS database used in our experiments, is a protein family database; compared with other protein databases, it focusses on general information on protein families. Our system found some relations that are particularly useful for PRINTS, such as is_a(protein, family), belong(protein, to:family), etc. under topic *function*. Other biologists might have different areas of interest or notions of what should relate to a given topic. We believe our system can be easily adapted to meet these needs; it requires neither pre-specified relations, nor hand-crafted rules or annotated corpora, just a collection of relevant and irrelevant sentences from MEDLINE for specific areas of interest.

Understandably, different performance levels were observed for different topics. While topic *structure* shows high precision and high recall, the other topics, *disease* and *function*, suffer from low recall. Considering that the sizes of *structure* and *function* corpora are similar, *function* seems to be a more difficult topic than *structure*. In a biomedical context, protein structure is about a protein itself and its components, and can be expressed in a simple way in texts, whereas function- and disease-related relations appear to exhibit a higher level of complexity and cannot be so narrowly defined. Clearly, more sophisticated methods are needed to handle difficult topics. Our comparative results on three topics show that performance is dependent on the topic complexity.

When *learning relations and IE rules from pre-labelled corpora*, we encountered the problem of evaluating an IE system, where relations to be extracted are not pre-defined. Typically, other IE methods test IE systems with gold-standard corpora where relations to be extracted have been specified. Our evaluation relied on the domain experts' judgment of whether extracted relations are correct or not.

7 CONCLUSION AND FUTURE WORK

The goal of our work is to alleviate the burden of developing IE systems for users who have little or no formal IE training. Our IE system allows them to build their own IE systems only with relevant and non-relevant sentences with regard to their interests.

Compared with document-level, sentence-level and keyword-level annotation, one of the advantages of relation-level annotation is the possibility of posing structured queries as demonstrated in (Karp, 2000). For example, with a set of extracted relations for a target protein (shown in Table 7), we can send a query like 'what biological entities can activate the target protein *NF-kappaB*?'

Learning to extract relations is a natural complement to previous and ongoing research on identifying named entities. Currently, many available named-entity taggers exist both for general domains (e.g. persons, organizations, locations) and specific domains (biology, chemistry, etc.), and their performance is very accurate. We believe that we can build on their results by applying our method to extract relations concerning named entities harvested by these taggers. To validate this, we are currently working on the NCI (National Cancer Institute) cancer gene data, which contain pairs of associated genes and cancers with supporting sentences from the biomedical literature.

Traditionally, IE suffers from low recall, especially when the complexity of a topic is high as shown in our results. We suppose that more IE rules should be learned for relations in difficult topics, and the size of training sets should be enlarged. In biomedical domains, there is an abundance of unlabelled texts that could be used for learning more IE rules. We propose to exploit these resources to increase recall without incurring additional costs in terms of manual labelling. Our next research plan is thus to combine labelled and unlabelled corpora using semi-supervised learning techniques.

ACKNOWLEDGEMENTS

The work reported in this article was funded by European Commission grant number QLRI-CT-2002-02770 BioMinT and the Swiss National Science Foundation in the framework of the ProDoTT project. We thank the CNTS team (University of Antwerp) for allowing us to use the MBSP parser.

Conflict of Interest: none declared.

REFERENCES

Attwood,T.K., *et al.* (2003) Prints and its automatic supplement, preprints. *Nucleic Acids Res*., **31**, 400–402.

Califf,M.E. and Mooney,R.J. (2003) Bottom-up relational learning of pattern matching rules for information extraction. *J. Mach. Learn. Res*., **4**, 177–210.

Collier,R. (1996) Automatic template creation for information extraction, an overview, Technical report, University of Sheffield.

Cootes,A. *et al*. (2003) The automatic discovery of structural principles describing protein fold space. *J Mol. Biol*, **330**, 839–850.

Daelemans,W., *et al*. (1999) Memory-based shallow parsing. In *Proceedings of CoNLL-99*.

Deng,M., *et al*. (2004) Mapping gene ontology to proteins based on protein-protein interaction data. *Bioinformatics*, **20**, 895–902.

Freitag,D. (2000) Machine learning for information extraction in informal domains. *Mach. Learn*., **39**, 169–202.

Gaizauskas,R.J., *et al*. (2003) Protein structures and information extraction from biological texts: The pasta system. *Bioinformatics*, **19**, 135–143.

Hasegawa,T., *et al*. (2004) Discovering relations among named entities from large corpora. In *ACL*, pp. 415–422.

Hu,Z.-Z., *et al*. (2005) Literature mining and database annotation of protein phosphorylation using a rule-based system. *Bioinformatics,* **21**, 2759–2765.

Karp,P.D. (2000) An ontology for biological function based on molecular interactions. *Bioinformatics*, **16**, 269–285.

Kim,J.-D., *et al*. (2003) Genia corpus – a semantically annotated corpus for bio-textmining. In *ISMB (Supplement of Bioinformatics)*, pp. 180–182.

Lavrac,N., *et al*. (2004) Subgroup discovery with cn2-sd. *J. Mach. Learn. Res*., 5, 153–188.

Lodhi,H. and Muggleton,S. (2004) Modelling metabolic pathways using stochastic logic programs-based ensemble methods. In *CMSB*, pp. 119–133.

Mitchell,A.L., *et al*. (2005) METIS: multiple extraction techniques for informative sentences. *Bioinformatics*, **21**, 4196–4197.

Muggleton,S. and Raedt,L.D. (1994) Inductive logic programming: theory and methods. *J. Logic Programming*, **19**, 629–679.

Nedellec,C. *et al*. (2001) Sentence filtering for information extraction in genomics, a classication problem. In *PKDD*, pp. 326–337.

Riloff,E. (1996) Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Articial Intelligence (AAAI- 96)*, pp. 1044–1049.

Saric,J., *et al*. (2006) Extraction of regulatory gene/protein networks from Medline. *Bioinformatics*, **22**, 645–650.

Soderland,S. (1999) Learning information extraction rules for semi-structured and free text. *Mach. Learn.*, *34*, 233–272.

Srinivasan,A. (2000) The Aleph manual. *Technical report*, Computing Laboratory, Oxford University.