Extraction of Genic Interactions with the Recursive Logical Theory of an Ontology

Alain-Pierre Manine, Erick Alphonse and Philippe Bessières

[1] Université Paris 13
LIPN-CNRS UMR7030
F93430 Villetaneuse
{manine,alphonse}@lipn.univ-paris13.fr
[2] Institut National de la Recherche Agronomique
MIG-INRA UR1077
F78352 Jouy-en-Josas
philippe.bessieres@jouy.inra.fr

Abstract. We introduce an Information Extraction (IE) system which uses the logical theory of an ontology as a generalisation of the typical information extraction patterns to extract biological interactions from text. This provides inferences capabilities beyond current approaches: first, our system is able to handle multiple relations; second, it allows to handle dependencies between relations, by deriving new relations from the previously extracted ones, and using inference at a semantic level; third, it addresses recursive or mutually recursive rules. In this context, automatically acquiring the resources of an IE system becomes an ontology learning task: terms, synonyms, conceptual hierarchy, relational hierarchy, and the logical theory of the ontology have to be acquired. We focus on the last point, as learning the logical theory of an ontology, and *a fortiori* of a recursive one, remains a seldom studied problem. We validate our approach by using a relational learning algorithm, which handles recursion, to learn a recursive logical theory from a text corpus on the bacterium *Bacillus subtilis*. This theory achieves a good recall and precision for the ten defined semantic relations, reaching a global recall of 67.7% and a precision of 75.5%, but more importantly, it captures complex mutually recursive interactions which were implicitly encoded in the ontology.

1 Introduction

The elucidation of molecular regulations between genes and proteins, as well as the associated physical interactions, is essential in the understanding of living organisms, as they underlie the control of biological functions. However, their knowledge is usually not available in formatted information from widely accessed international databanks, but scattered in the unstructured texts of scientific publications.

For this reason, numerous works in recent years have been carried out to design Information Extraction (IE) systems, which aim at automatically extracting genic interaction networks from bibliography (see e.g. [1] for a review).

To perform extraction, a possible method is to start with a model of the domain, i.e. an ontology, which defines concepts (e.g. gene, protein) and an interaction relation [2]. Then, an *ontology population* procedure is achieved [3]: concepts and relations mentioned in the text are recognized and instantiated. To do so, after a preliminary *terms* and *named entities recognition* step, which leads to the instantiation of main concepts, semantic relations are usually extracted by applying so-called *extraction patterns*, or *rules*. For instance, in the following sentence:

> Production of sigmaK about 1 h earlier than normal does affect Spo0A [...]

the protein concepts are first instantiated (sigmaK, Spo0A); subsequently, an interaction relation is instantiated between the sigmaK and Spo0A proteins. Rules applied to identify the former relation exhibit syntactico-semantic features (e.g., syntactic relations between sigmaK and Spo0A words) originated from NLP (Natural Language Processing) modules.

Designing rules in order to capture the relevant knowledge underlying the concept of *genic interaction* is a very difficult challenge, as this concept covers a wide variety of interdependent phenomenons (protein and gene regulations, DNA binding, phosphorylation, etc.). For instance, the previous example implies an unspecified regulation (sigmaK is only stated to "affect" Spo0A), whereas in the following sentence:

> Here, we show that GerE binds near the sigK transcriptional start site [...]

something very specific, a physical binding between the GerE protein and a DNA site, is described; furthermore, a more generic relation, an interaction between GerE and sigK, can be deduced from this binding, on which it depends. Despite this variety of relations, and their interrelations, most rules of IE systems are limited to extract a unique type of interaction relation. Consequently, they face a trade-off between recall and precision. Some favour precision by focusing on very specific and well-defined interactions, like protein-protein interactions (e.g. [4–8]), but neglect other biological phenomenons; whereas other stress on recall by extracting general relations (e.g. [9, 10]), but face precision issues originating from the important lexical diversity.

To overcome this trade-off and to be able to model more accurately the biological field, IE systems require more expressive extraction rules. Firstly, it is not sufficient to address one single interaction relation: rules have to involve multiple relations, defined within an arbitrarily complex ontology [3], in order to model, for instance, that GerE *binds to* (first relation) a site *included in* (second relation) the sigK gene. Secondly, syntactico-semantic rules alone are inadequate. Semantic reasoning is needed to express semantic relations dependencies, and to deduce, for instance, that if GerE *binds to* a site *included in* sigK, then GerE *interacts with* sigK. Such a reasoning requires to be able to infer new relations (*interacts with*) from the previously instantiated ones (*binds to*, *included in*),

something beyond the inference capabilities of the current approaches. Thirdly, recursive or mutually recursive rules have to be handled; recursion is indeed intrinsic to natural language (see, for instance, [11, 12]), as illustrated by the transitive nature of several relations: if the DNA site A is *included in* another site B itself *included in* C, then A is *included in* C.

We propose an integrated approach to address these three points, in which the logical theory of an ontology generalises regular IE patterns and is responsible for the extraction. We denote by *ontology* both a conceptual and a relational hierarchy (the thesaurus), along with a logical theory (see e.g. [13]), which expresses constraints and dependences between concepts.

The logical theory is able to refer to any concept defined in the ontology, and as such, to handle multiple inter-dependent relations, in accordance with our first point; these dependences may be recursive, in agreement with the third. Furthermore, ontologies exhibit inference capabilities of current knowledge representation languages, like OWL(-DL), Flogic or Datalog (see e.g. [14, 15]), which allow to achieve semantic reasoning, as required by the second point. For instance, semantic knowledge may be expressed in Datalog by the following type of rules of the logical theory:

$$interact(A, B) \leftarrow bind\_to(A, C), included\_in(C, B), \\ protein(A), gene(B), dna\_site(C)$$

which means that: "A interacts with B, if A binds to a DNA site C, which is included in the gene B".

Extraction rules may be crafted by the domain expert as part as background knowledge, or automatically learnt with machine learning techniques. We choose the latter alternative, which has been well-motivated in IE as a generic component easily adaptable to new domains [16, 17]. In our context, rule acquisition becomes part of an ontology learning task: terms, synonyms, the conceptual hierarchy (e.g. [18]), the relational hierarchy (e.g. [2]) and the logical theory of the ontology have to be learnt from a domain corpus. We focus on this latter point which has been seldom addressed, although it is an important prerequisite to complex knowledge-based systems, and we used the multiple predicate learning system Atre [19] to produce recursive rules with the suitable expressiveness. This work extends our previous work [3] in several ways. First, it motivates the use of the logical theory of the ontology as a proper generalization of the extraction rules or patterns. Second, neither relation dependencies nor recursion were taken into account during learning, which limited the expressiveness of the IE system: this is the key element that allows to conduct semantic reasoning and derive new relations from previously extracted ones. Finally, the corpus has been enriched with recursive and interdependent biological phenomena not processed previously, and is made publicly available[3].

The plan of the article is as follows. We discuss related works on IE and machine learning in section 2. We recall our ontology-population based IE plateform in section 3. We present our ontology learning strategy in section 4. In section 5, we report and comment our learning results on the bacterium corpus. Finally,

[3] `http://www-lipn.univ-paris13.fr/~alphonse/IE/genic_interaction`

in section 6, we discuss our approach and propose some perspectives in IE from text.

2 Related works

Whereas we aim at automatically acquiring inference rules of an ontology, [20] notes that, in the ontology learning field, very few works are related to this task, as most researches focus on taxonomy and non-hierarchical relations learning. Work of [21] is loosely connected to it, as they learn simple association rules to handle paraphrases; more recently, [22] focus on learning non-domain-specific rules, like inclusion or disjointness statements between concepts, while we acquire domain-specific relations, like binding or regulatory interaction.

These rules cannot be acquired by machine learning techniques usually exploited to learn IE extraction patterns. Binary classification is indeed mostly used (e.g. [23, 16]), and is limited to learn a single relation, whereas we need multiple conceptual relations. Furthermore, if multi-class learning is occasionally involved [4, 24], this strategy does not yield to the required expressivity level, as they assume independence between target predicates, which forbids recursion. In the same way, a multi-class algorithm is used in [3], which only learns non-recursive syntactico-semantic patterns: in contrast to our approach, recursive clauses or rules based on *previously deduced relations* are not learnable. It was proposed to make use of stratified learning where recursive phenomena were identified, isolated and left to the expert: recursive rules were manually input during the ontology design. However, this approach does not scale well as it is too difficult to identify in the text those phenomena, implicit in the ontology.

To be able to produce recursive or mutually recursive target predicates, we chose to take advantage of a relational learning algorithm in the multiple predicate setting. To the best of our knowledge, the only other IE application of multiple predicate learning is found in [25], but is limited to named entities recognition, whereas we focus on extracting relations between recognized entities.

3 Information Extraction platform

Our information extraction platform architecture is presented in figure 1. During production (right of the figure), it involves two main stages: firstly, a preliminary ontology population step, during which outputs of NLP modules are normalized in the ontology language by an *ontology population module*, and secondly, inference made by a *query module* based on the logical theory of the ontology in order to derive new instances.

3.1 Ontology Population Module

The first phase, the ontology population, is the extraction from text of instances of concepts and relations defined in the ontology. As it requires complex mappings between expressions in natural language to ontology structures [26], going

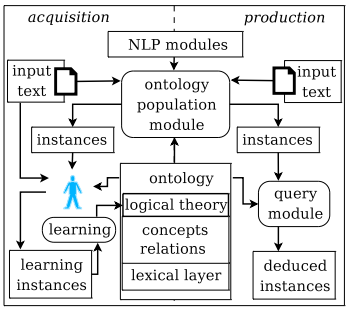

Fig. 1. Ontology-based IE platform.

beyond mere class/label linking (like the `rdf:label` property of RDF[4], or the more complex properties of SKOS[5], some authors introduce lexicon models [26, 27] to ground the semantic information to the linguistic domain, even though they do not employ it in an IE context. We follow this approach by providing a so-called *Lexical Layer* (LL) along with the ontology. However, where the previous authors follow a linguistic point of view, by proposing a model to link ontology structures to lexical descriptions, we adopt an application-oriented perspective. Our LL is a task-dependent parameter: it comprises classes and relations required to link the output of NLP modules to the ontology, so it is designed with respect to those NLP modules. Its purpose is to provide a representation with sufficient expressiveness for efficient inference. These classes and relations define normalizations of text in intermediate stages of abstraction, between raw text and conceptual level. For instance, a LL relation may associate a syntactic label with an instance, or a syntactic relation between two instances (subject (*"subj"*) and object (*"obj"*) relations in figure 3). The LL is described in the same language as the ontology, so the inference rules can benefit from it.

Figure 2, in plain lines, exemplifies an output of the ontology population module. Instances of the *protein* concept (GerE, sigmaK) have been instantiated by a terminological module. They have been properly linked with existing domain knowledge, through the *product_of* semantic relation, which states that the protein sigmaK is encoded by the sigK gene. Subject (*subj*), object (*obj*), *comp_from*, and *comp_by* relations belong to the lexical layer, and their instantiations originate from a parser. A fragment of the corresponding ontology is shown in figure 3. Dashed lines exemplify the declarative definition of the lexical layer (e.g. *subj*, *obj*). "stimulate" is an instance of the concept *regulation*, and "use" is an instance of the *dependence* concept. Both are required to understand the presence of a regulation between proteins, and were thus added to the lexical layer. A transcription event occurs from (*t_from*) a promoter, and results from the action (*t_by*) of a protein. Therefore, promoters may be dependent (*p_dep*) of

[4] http://www.w3.org/TR/rdf-schema/
[5] http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102

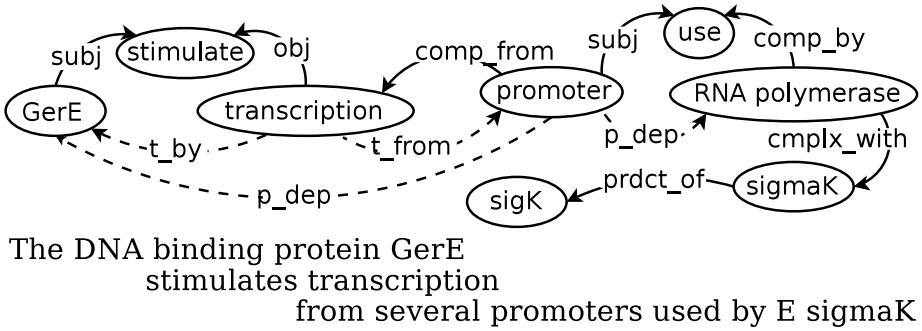

Fig. 2. Ontology Population output (plain lines), and some relations derived from the logical theory (dashed lines).

proteins. Finally, a protein complex results from the assembly of several proteins (*"complex_with"*): the protein complex EsigmaK is formed by a RNA polymerase complexed with the protein sigma K.

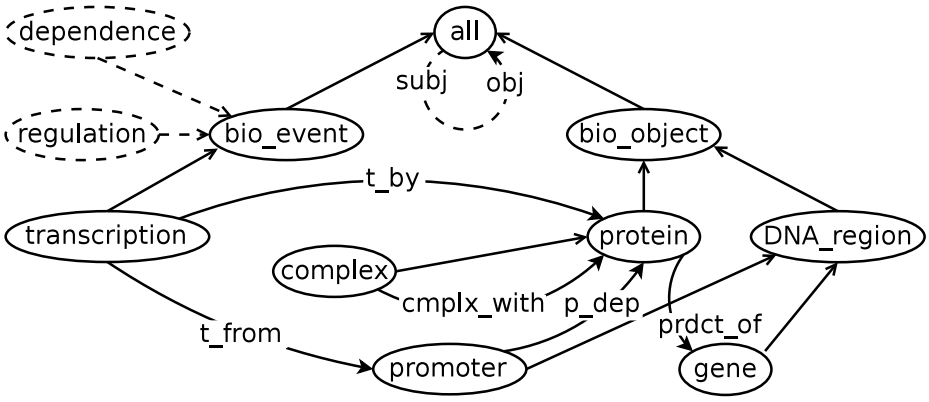

Fig. 3. Fragment of an ontology of biological interactions (lexical layer in dashed lines).

3.2 Query Module

The output of the ontology population module results from NLP modules and from domain knowledge (the latter allows us to know, for instance, that the sigma K protein is the product of the sigK gene, see figure 2). In opposition to traditional IE systems in which new facts are extensively extracted, here, knowledge is intensively encoded into the ontology structure, both within the conceptual hierarchy and within the logical theory, and is available through the

mean of user queries. To benefit from the inference capabilities of our system, the logical theory of the ontology is used to derive more instances from those previously extracted. This is done through our *query module*. Figure 2, in dashed lines, exemplifies such deduced instances. Consider the following user query, related to the sentence in figure 2:

```
?- p_dep(A,B).
```

which means "is there a promoter A dependent on a protein B?". An answer of the query module will be:

```
A = promoter,
B = GerE
```

The promoter was inferred to be dependent on the GerE protein thanks to the logical theory of the ontology, encoded as a clausal theory written in Datalog. The following rules was used:

$p\_dep(A, B) \leftarrow t\_by(C, B), t\_from(C, A)$

It means that "if a transcription event C is due to a protein B and occurs from a promoter A, then A is dependent on B". In the example, the *promoter dependence* relations between *promoter* and *GerE* (*p_dep(promoter, GerE)*) is true as both the relevant *transcription by* (*t_by(transcription, GerE)*) and *transcription from* relations (*t_from(transcription, promoter)*) are true.

Note that the former rule involves semantic attributes, whereas the other dashed relations have been deduced from syntactico-semantic inference, based on features belonging to the lexical layer.

The *transciption by* relation (*t_by*) was deduced from a rule like this:

$t\_by(A, B) \leftarrow subj(A, C), obj(B, C),$
$regulation(C), transcription(A), protein(B)$

which asserts that the transcription A is caused by protein B, if A is subject of a regulation event C, and if an object relation links B to C. In figure 2, *t_by(transcription, GerE)* is true, as *subj(GerE, stimulate)* is true and *obj(transcription, stimulate)* is true.

The *transcription from* relation (*t_from*) was inferred from the following type of rule:

$t\_from(A, B) \leftarrow comp\_from(B, A),$
$transcription(B), promoter(A)$

which asserts that the syntactic relation *comp_from* has the semantic value of a *transcription from* relation if the arguments of the relation are respectively a transcription instance and a promoter instance. In the figure, *t_from(transcription, promoter)* is true as *comp_from(promoter, transcription)* is true.

The previous examples illustrate how rules of the logical theory form a major part of our system; the next section describes the approach that allowed to automatically acquire them.

4 Learning the Logical Theory of the Ontology

As opposed to previous approaches (see section 2), learning takes place in the ontology language to produce a logical theory that holds true in the domain ontology and the lexical layer. From a machine learning point of view, the learner uses the ontology as the hypothesis language and instantiations of the ontology as the example language. During the acquisition of the theory, as illustrated in figure 1 (left part), the domain expert has to provide learning examples defined as instantiations of the ontology. He creates instances of concepts and relations of the ontology from a corpus, some instances being output by the ontology population module. Figure 2 exemplifies such annotation, the dashed lines corresponding to relations to learn.

Learning from such a relational language is known as Inductive Logic Programming (ILP) [28], where the hypothesis and the example languages are subsets of first-order logic. We encode the logical theory as a clausal theory, in Datalog. This is a knowledge representation language expressive enough for the task (as expressive as multi-relational databases), and theoretically well-understood in ILP, that most learners handle as learning language.

To learn from this relational language, we used the ATRE system [19], which handle recursive logical theories. A definition of a recursive theory, founded on the notion of *dependency graph*, is given by [19]. The dependency graph of a theory T is a directed graph $\gamma(T) = \langle N, E \rangle$, in which (i) each predicate of T is a node in N and (ii) there is an arc in E directed from a node a to a node b, iff there exists a clause C in T, such that a and b are the predicates of a literal occurring in the head and in the body of C, respectively.

This notion makes easier the characterization of multiple predicate learning relatively to multi-class learning: the dependency graph of a theory learned in the multi-class ILP setting will only comprise nodes, whereas in the multiple predicate case, it will include nodes and edges. Multiple predicate ILP may allow to learn recursive theory, i.e. a theory T where $\gamma(T)$ will contain at least one cycle.

The main problem to learn such a theory is related to the non-monotonicity property of the normal ILP setting [19]. In normal ILP setting, theories are induced thanks to a *separate-and-conquer* strategy: clauses are learnt one by one, covered examples are removed from the training set, and the process iterates until no more positive examples remained; in the multiple predicates paradigm, whenever two individual clauses are consistent in the data, their conjunction need not be consistent in the same data. ATRE addresses these issues by generating clauses all together, using a *separate-and-parallel-conquer* strategy.

ATRE represents examples as ground multiple-head clauses, called *objects*, which have a conjunction of literals in the head (because of space requirements, we refer the reader to [19] to an extensive description of ATRE). In our case, each sentence matches an object, and negatives examples were generated using a

closed-world assumption. For instance, the previous example will be equivalently represented as[6]:

$$t\_by(id2, id1), p\_dep(id4, id1), t\_from(id2, id4),$$
$$\neg t\_by(id1, id2), \neg t\_by(id1, id3), [\ldots] \leftarrow$$
$$subj(id1, id3), obj(id2, id3), comp\_from(id4, id2),$$
$$transcription(id2), protein(id1),$$
$$regulation(id3), promoter(id4).$$

Note that all the ontological knowledge is given as background knowledge to the ILP algorithm, like the generalisation relation between concepts. For instance, specifying that a protein complex is a protein etc. will be represented as a clausal theory:

$$protein(A) \leftarrow protein\_complex(A).$$
$$gene\_product(A) \leftarrow protein(A).$$
$$gene\_product(A) \leftarrow rna(A).$$

Processing an example involving a protein complex or a RNA, the learning algorithm chooses the most relevant generality level (e.g. "protein complex", "protein" or "gene product") to learn the logical theory.

5 Results

As previously stated, extracting a regulation network in other works is mostly restricted to the extraction of a unique binary interaction relation. Consistently, recent trends regarding the application of machine learning to biological IE head toward the development of public annotated corpora, targeting such binary relations to compare systems' performances (e.g. AIMed [29], Bioinfer [30], HPRD50 [10], LLL [9]). In this paper, the ontology does not limit us to the extraction of a single relation, but allows the definition of numerous relations. We present a way to encode extraction patterns in order to infer new knowledge from them. Seemingly, public corpora are inadequate to validate the inference capabilities of the logical theory, as well as the relevance of multiple predicate ILP to acquire it.

We used the ontology of gene transcription in bacteria introduced in [3]. It describes the structural model of a gene, its transcription, and associated regulations, to which biologists implicitly refer in their texts. The ontology includes some forty concepts, mainly about biological objects (gene, promoter, binding site, RNA, operon, protein, protein complex, gene and protein families, etc.), and biological events (transcription, expression, regulation, binding, etc.). We focus on the ten defined conceptual relations: a general, unspecified, interaction relation (i), and nine relations specific to some aspects of the transcription (binding, regulons and promoters). The specific relations are the following: promoter dependence (p_dep), promoter of (p_of), bind to (b_to), site of (s_of),

[6] Some negative examples have been omitted.

Name	Example
p_dep	*sigmaA* recognizes **promoter elements**
p_of	the *araE* **promoter**
b_to	**GerE** binds near the sigK *transcriptional start site*
s_of	*-35 sequence* of the **promoter**
rm	*yvyD* is a member of sigmaB **regulon**
r_dep	*sigmaB* **regulon**
t_from	**transcription** from the Spo0A-dependent *promoter*
t_by	**transcription** by final *sigma(A)-RNA polymerase*
et	**expression** of *yvyD*
i	**KinC** was responsible for Spo0A~P *production*

Table 1. (From [3]) List of relations defined in the ontology, and phrase examples (sub-terms of the relation are shown in italic and bold).

regulon member (*rm*), regulon dependence (*r_dep*), transcription from (*t_from*), transcription by (*t_by*), event target (*et*). As an illustration of their semantics, table 1 gives, for each relation, an expression where the relation is needed to normalise it. For instance, the third line in the table states that, in the sentence "GerE binds near the sigK transcriptional start site", the protein "GerE" (in bold font) binds to (*b_to*) the site "transcriptional start site" (in italics).

The lexical layer encompasses syntactic relations between classes, and syntactico-semantic classes aimed at factorizing entities, which may share the same syntactical context (gene and protein, gene family and protein family, transcription and expression events).

We validate the interest of multiple predicate ILP in an ontology learning context by reusing the corpus presented in [3]. This corpus is a reannotation of the LLL corpus [9]: 160 sentences, provided with dependency-like parsing with resolved coreferences, have been reannoted with terms, concepts and relations according to the ontology. This corpus have been curated and augmented with new relations that were left out in [3] because they were matching expert rules with recursion or dependencies with other rules. In total, 711 relations were available for learning.

We used a ten-fold cross-validation to evaluate recall and precision of the IE process. In order to evaluate the gain of recursive rules, we ran Atre with and without recursive learning enabled. The results are shown in table 2 and table 3, respectively. Although recursion allows to model more complex interactions, it is interesting to note that the recursive theory also yields better results on this corpus, with a global recall of 67.7%, compared to 65.6%, and a precision of 75.5%, compared to 71.7%. The scores are satisfactory, and corroborate the relevance of our ontology learning approach. More specific relations (*et*, *t_from*, *r_dep*) have little lexical variability, and reach high scores; on the contrary, more general ones, like *i*, exhibiting greater variability, are noticeably harder to learn. The poor score of *rm* may be due to an unbalanced distribution of this relation through Atre's objects.

In the following, we will illustrate the benefit of the multiple predicate learning paradigm by outlining a typology of the learned rules. First of all, some rules only exhibit semantic attributes, allowing to exclusively reason on a semantic level.

$$i(X2, X1) \leftarrow t\_by(X2, X3), et(X3, X1). \quad (1)$$
$$s\_of(X2, X1) \leftarrow t\_from(X3, X2), et(X3, X1). \quad (2)$$

For instance, (1) expresses that if X1 is transcribed by X2, then they interact (e.g. "gspA" and "sigma B" in "transcription of gspA is sigma B dependent"); (2) asserts that if the X1 gene is transcribed from the X2 promoter, then X2 is a site included in X1 (e.g. "spoVD" and "promoter" in "spoVD transcription appears to occur from a promoter").

Relation	Recall (%)	Prec. (%)	Number
i	50.2	70.6	225
rm	33.3	41.7	15
r_dep	100.0	100.0	12
b_to	69.6	75.3	79
p_dep	69.8	71.2	53
s_of	61.2	61.2	67
p_of	69.8	55.6	43
et	95.7	96.9	164
t_from	73.3	84.6	15
t_by	52.6	62.5	38
Global	67.7	75.5	711

Table 2. Results for multiple predicate learning (with recursion). Last column shows the number of examples.

Relation	Recall (%)	Prec. (%)	Number
i	57.3	74.5	225
rm	33.3	62.5	15
r_dep	100.0	100.0	12
b_to	67.0	72.6	79
p_dep	67.9	61.0	53
s_of	73.1	54.4	67
p_of	69.7	44.1	43
et	76.8	96.1	164
t_from	60.0	81.8	15
t_by	47.3	69.2	38
Global	65.6	71.7	711

Table 3. Results for multi-class learning (without recursion). Last column shows the number of examples.

Multiple predicate setting is especially well-fitted to the hierarchical structure of ontologies:

$$s\_of(X2, X1) \leftarrow p\_of(X2, X1). \quad (3)$$
$$p\_of(X2, X1) \leftarrow s\_of(X2, X1), promoter(X2), gene\_entity(X1). \quad (4)$$

Rule (3), given by the expert as domain knowledge, encodes an *is-a* relation between *p_of* and *s_of*, whereas learned rule (4) allows to specialise a *s_of* relation into a *p_of* relation, if X2 is a promoter and X1 a gene. This is illustrated by the last example of the previous paragraph: thanks to (2) and (4), the system will deduce a *p_of* relation between the promoter and the spoVD gene. Note that the rules (2), (3), (4) constitute a recursive theory.

Previous kind of rules are grounded to NL through predicates that involve LL-defined literals (i.e. syntactico-semantic attributes), like:

$$i(X2, X1) \leftarrow subj\_v\_n(X3, X1), obj\_v\_n(X3, X2), term(X3, require). \quad (5)$$
$$i(X2, X1) \leftarrow subj\_v\_n(X3, X2), obj\_v\_n(X3, X1), regulation(X3). \quad (6)$$

Rules (5) and (6) allow to derive semantic relations from syntactic relations. (5) is related to expressions like "A activates B", while (6) handles phrases like "B requires A" (note the argument order). These two rules show that Atre is able to learn classes of terms not explicitly defined by the expert to derive the argument order.

Our approach has the capacity to combine various abstraction levels in order to deduce new relations. For instance, the recursive rule (7) expresses that if protein X2 binds to (semantic level relation) site X3, included in (semantic level relation) site X4, then a *comp_n_n_of* (syntactic level relation) between X4 and X1 implies that X2 binds to X1 (e.g. "GerE" and "promoter" in "GerE binds to two sites that span the -35 region of the cotD promoter"). Previously inferred semantic relations may also be useful as contextual disambiguation clues. In (8), the *et* relation ensures that a *comp_v_pass_n_from* syntactic relation has the semantic value of a *t_from*.

$$b\_to(X2, X1) \leftarrow b\_to(X2, X3), s\_of(X3, X4), comp\_n\_n\_of(X4, X1). \quad (7)$$
$$t\_from(X2, X1) \leftarrow et(X2, X3), comp\_v\_pass\_n\_from(X2, X1). \quad (8)$$

Moreover, reasoning on multiple abstraction levels allows to factorize various lexical variations into a single semantic label. As a result, the learner produces more compact theories. Rule (9) clarifies this point. It will match expressions either like "the cwlB operon is transcribed by E sigma D" or like "transcription of cotD by sigmaK RNA polymerase", as the two forms "transcription of A" and "A is transcribed" are factorized by rules (10) and (11). In the multi-class ILP setting, two rules would have been required.

$$i(X2, X1) \leftarrow comp\_n\_n\_by(X3, X2), \quad (9)$$
$$et(X3, X1).$$
$$et(X2, X1) \leftarrow comp\_n\_n\_of(X2, X1), \quad (10)$$
$$event(X2).$$
$$et(X2, X1) \leftarrow subj\_v\_pass\_n(X2, X1), \quad (11)$$
$$transcription(X2).$$

6 Conclusion and perspectives

Automatic extraction of genetic pathways from scientific litterature involves the modelling of a wide variety of semantic relations that are intrinsically interrelated. However, interrelations are neglected by traditional IE approaches, which only focus on the mapping of syntactico-semantic structures and semantic relations, and assume independence between semantic relations. In this paper, we introduced an IE platform that overcomes these limitations and exhibits inference capabilities going beyond existing systems by generalizing traditional IE patterns with the logical theory of an ontology. In particular, it allows to define multiple relations and to derive new relations from previously instantiated ones, when the former depend on the latter. Dependencies and recursive dependencies required by the logical theory are learnt from an annotated corpus by taking advantage of ILP in the multiple predicate setting, using the Atre system, which does not suffer from the independence assumption of usual machine learning approaches. We validated our system by learning a recursive logic theory from a bacterium corpus, and discussed its relevance for IE, especially its capacity to combine syntactic and semantic reasoning, and to benefit from the hierarchical structure of the ontology (specialisation and generalisation rules).

In the future, the declarative nature of our platform will allow its easy extension. Specifically, we plan to handle regulations, like inhibition and activation relations, a very important demand from biologists yet to be fulfilled. It may be due to the fact that these relations are inherently mutually recursive: only when we know that A inhibits B, which in turn inhibits C, that we can derive that A activates (or participates in the activation of) C.

Furthermore, we plan to survey the capacity of ILP tools, learning in the multiple predicate setting, to scale up and to handle noise, as this is a crucial requirement for NLP applications.

References

1. Ananiadou, S., Kell, D.B., Tsujii, J.: Text mining and its potential applications in systems biology. Trends in Biotechnology **24** (2006)
2. Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., Rojas, I.: Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI2005), Edinburgh, UK (2005)

3. Manine, A.P., Alphonse, E., Bessiere, P.: Information extraction as an ontology population task and its application to genic interactions. In: 20th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2008. Volume 2. (2008) 74–81
4. Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, AAAI Press (1999) 77–86
5. Rindflesch, T., Tanabe, L., Weinstein, J., Hunter, L.: EDGAR: extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the Fifth Pacific Symposium on Biocomputing (PSB'03). (2000) 517–528
6. Blaschke, C., Andrade, M., Ouzounis, C., Valencia, A.: Automatic extraction of biological information from scientific text: Protein-protein interactions. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, AAAI Press (1999) 60–67
7. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T.: Automated extraction of information on protein-protein interactions from the biological literature. Bioinformatics **17** (2001) 155–161
8. Saric, J., Jensen, L., Ouzounova, R., Rojas, I., Bork, P.: Large-scale extraction of protein/gene relations for model organisms. In: First International Symposium on Semantic Mining in Biomedicine 2005. (2005)
9. Nédellec, C.: Learning language in logic — Genic interaction extraction challenge. In Cussens, J., Nédellec, C., eds.: Proceedings of the Fourth Learning Language in Logic Workshop (LLL05). (2005) 31–37
10. Fundel, K., Küffner, R., Zimmer, R.: RelEx — relation extraction using dependency parse trees. Bioinformatics **23** (2007) 365–371
11. Hauser, M.D., Chomsky, N., Fitch, W.T.: The faculty of language: What is it, who has it, and how did it evolve? Science **298** (2002) 1569–1579
12. Bostrm, H.: Induction of recursive transfer rules. In: Proc. of Learning Language in Logic (LLL) Workshop, Springer-Verlag (2000) 52–62
13. Gómez-Pérez, A.: Ontological engineering: A state of the art. Expert Update **2** (1999) 33–43
14. McGuinness, D., van Harmelen, F.: OWL web ontology language overview: W3C recommendation 10 february 2004. Technical report, W3C (2004)
15. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. J. ACM **42** (1995) 741–843
16. Salakoski, T., Rebholz-Schuhmann, D., Pyysalo, S., eds.: Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland, Turku Centre for Computer Science (TUCS) (2008)
17. Krallinger, M., Leitner, F., Valencia, A.: Assessment of the second BioCreAtIvE PPI task: Automatic extraction of protein-protein interactions. In: Proceedings of the Second BioCreAtIvE Challenge Evaluation Workshop. (2007) 41–54
18. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence Research (JAIR) **24** (2005)
19. Varlaro, A., Berardi, M., Malerba, D.: Learning recursive theories with the separate-and-parallel conquer strategy. In: Proceedings of the Workshop on Advances in Inductive Rule Learning in conjunction with ECML/PKDD. (2004) 179–193
20. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: An overview. In Buitelaar, P., Cimiano, P., Magnini, B., eds.: Ontology Learning from Text:

Methods, Evaluation and Applications. Volume 123 of Frontiers in Artificial Intelligence and Applications. IOS Press (2005)
21. Lin, D., Pantel, P.: DIRT discovery of inference rules from text. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2001) 323–328
22. Vlker, J., Vrandecic, D., Sure, Y., Hotho, A.: Learning disjointness. In Franconi, E., Kifer, M., May, W., eds.: Proceedings of the 4th European Semantic Web Conference (ESWC'07). Volume 4519 of Lecture Notes in Computer Science., Springer (2007) 175–189
23. Riloff, E.: Automatically generating extraction patterns from untagged text. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), AAAI Press / The MIT Press (1996) 1044–1049
24. Rosario, B., Hearst, M.A.: Classifying semantic relations in bioscience texts. In: ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2004) 430
25. Berardi, M., Malerba, D.: Learning recursive patterns for biomedical information extraction. In Muggleton, S., Otero, R.P., Tamaddoni-Nezhad, A., eds.: ILP. Volume 4455 of Lecture Notes in Computer Science., Springer (2006) 79–93
26. Cimiano, P., Haase, P., Herold, M., Mantel, M., Buitelaar, P.: LexOnto: A model for ontology lexicons for ontology-based NLP. In: Proceedings of the OntoLex07 Workshop held in conjunction with ISWC'07. (2007)
27. Buitelaar, P., Sintek, M., Kiesel, M.: A multilingual/multimedia lexicon model for ontologies. In Sure, Y., Domingue, J., eds.: ESWC. Volume 4011 of Lecture Notes in Computer Science., Springer-Verlag (2006) 502–513
28. Muggleton, S., Raedt, L.D.: Inductive Logic Programming: Theory and methods. Journal of Logic Programming **19,20** (1994) 629–679
29. Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K., Wong, Y.W.: Comparative experiments on learning information extractors for proteins and their interactions. Artificial Intelligence in Medicine **33** (2005) 139–155
30. Pyysalo, S., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics **9** (2008) S6