Information Extraction as an Ontology Population Task and its Application to Genic Interactions

Alain-Pierre Manine, Erick Alphonse
Université Paris 13, Institut Galilée
Laboratoire d'Informatique Paris-Nord
LIPN, Univ. Paris13/CNRS UMR7030
99 ave. J.-B. Clément, F93430 Villetaneuse
{manine,alphonse}@lipn.univ-paris13.fr

Philippe Bessières
Institut National de la Recherche Agronomique
Unité Mathématique, Informatique et Génome
MIG, INRA UR1077
F78352 Jouy-en-Josas
philippe.bessieres@jouy.inra.fr

Abstract

Ontologies are a well-motivated formal representation to model knowledge needed to extract and encode data from text. Yet, their tight integration with Information Extraction (IE) systems is still a research issue, a fortiori with complex ones that go beyond hierarchies. In this paper, we introduce an original architecture where IE is specified by designing an ontology, and the extraction process is seen as an Ontology Population (OP) task. Concepts and relations of the ontology define a normalized text representation. As their abstraction level is irrelevant for text extraction, we introduced a Lexical Layer (LL) along with the ontology, i.e. relations and classes at an intermediate level of normalization between raw text and concepts. On the contrary to previous IE systems, the extraction process only involves normalizing the outputs of Natural Language Processing (NLP) modules with instances of the ontology and the LL. All the remaining reasoning is left to a query module, which uses the inference rules of the ontology to derive new instances by deduction. In this context, these inference rules subsume classical extraction rules or patterns by providing access to appropriate abstraction level and domain knowledge. To acquire those rules, we adopt an Ontology Learning (OL) perspective, and automatically acquire the inference rules with relational Machine Learning (ML). Our approach is validated on a genic interaction extraction task from a Bacillus subtilis *bacterium text corpus. We reach a global recall of 89.3% and a precision of 89.6%, with high scores for the ten conceptual relations in the ontology.*

1 Introduction

Emerging Information Extraction (IE) applications, like extracting biological pathways from bibliographical databanks [29], require elaborated representations of the extracted results. Ontologies are an adequate formal representation able to convey such a complex knowledge, hence they become preeminent in the IE field, but most authors exploit it punctually. Their structure may be used as a basis to craft extraction rules [33, 15], or as a useful disambiguation resource. For instance, [11, 16] apply it to solve coreferences; [13] select relevant syntactic graphs from a parser using the structure of an ontology; [33] discuss the benefit of an ontology to solve some syntactical ambiguities relying on concepts arity. In most IE pipelines, ontology (often limited to a conceptual hierarchy) is only used to enrich the text with semantic categories. Information extraction is then delegated to the construction of so-called extraction patterns or rules, applied on their output (e.g. [19, 2, 13, 25]). Some authors have stressed the benefit to also use the ontology as the IE output language, in the context of Ontology Population (OP) [18, 21]. IE and OP share the same goal, namely to enrich a knowledge base with new instances specified in the ontology. Yet, ontology integration into an IE platform, beyond conceptual hierarchies, is still a research issue.

Here, we present an original architecture that fully integrates ontologies into the IE process. We illustrate its application to genic interaction extraction from scientific literature, where a rich modeling of the domain allows to extract information beyond the capabilities of current systems.

We refer to an ontology as a thesaurus (concept and relation hierarchies), along with a logical theory given as a set of inference rules (see e.g. [17]). In our approach, an IE system is fully specified by designing an ontology, and its conception is an OP task. In this context, the inference rules of the ontology subsume classical extraction rules or patterns by providing access to appropriate abstraction level, domain knowledge, and semantic inference capabilities of current knowledge representation languages, like OWL(-DL) or Flogic [24, 20]. Inference rules reasoning on

the ontology are able to gather information otherwise scattered throughout bibliographical databases, and to discover knowledge not explicitly stated in texts.

The ontology is seen as a specification of a decontextualized and normalized representation of the text, where an OP module processes the output of NLP modules, to produce relevant instantiations. However, as written by [14], ontologies are not necessarily useful to IE, in the sense that the granularity of the classes between a conceptual and a sublanguage model may differ. To bridge the gap between expressions in Natural Language (NL) and ontology structures, some authors introduce lexicon models [12, 10], to ground the semantic information to the linguistic domain. We follow this approach and introduce, along with the ontology, what we call a Lexical Layer (LL), i.e. relations and classes at an intermediate level of abstraction between raw text and concepts. As opposed to related IE works, the extraction itself, being done by the OP module, only normalizes terms and simplest lexical variations, in accordance to the domain ontology and the LL. More complex variations are handled by a query module, which derives new instances from the inference rules of the ontology, which may be deduced from the syntactico-semantic level using the LL, or from the semantic level.

Inference rules may be crafted by the domain expert as part of the ontology design, or automatically learnt by Machine Learning (ML) techniques. We focus on this latter case, which is considered in the context of IE systems as a generic component to easily adapt them to new domains. However, as opposed to previous approaches, learning takes place in the ontology language, to produce deductive rules which hold in the domain ontology. From a ML point of view, the learner uses the ontology as hypothesis language, and instantiations of the ontology as example language. In our approach, this casts the important work on extraction pattern learning into the Ontology Learning (OL) paradigm [8], which has almost not been explored in IE.

We discuss related works on OP and OL in section 2, and present our IE system architecture in 3. In section 4, we validated our architecture by designing an ontology to model genic interactions, and annotating a corpus of gene transcription in *Bacillus subtilis*, in order to learn relevant inference rules of the ontology. Answer examples from the query module illustrate the benefit of reasoning on the ontology model to present instances to the user. Finally, in section 5, we discuss our approach and propose some perspectives in IE from text.

2 Related works

Broadly speaking, in the general domain, OP systems (like [18, 1, 21, 9]) are related to mono-slot extraction and Named Entities Recognition (NER), whereas we are mostly interested in multi-slot or *relational* IE, i.e. extracting relations between previously recognized entities. Furthermore, most systems solely use ontologies as an output format, and few employ them as a resource during the extraction process. We can mention [31, 21], who take into account conceptual relations to improve NER.

Automatic acquisition of inference rules of the ontology is considered as part of OL. However, as noted in [8], very few works are related to this approach, focusing more on taxonomy and non-hierarchical relation learning (e.g. [23]). Works of [22] are loosely related to our task, learning simple association rules to handle paraphrases, and more recently, [34] focus on learning non-domain-specific rules, like inclusion or disjointness statements between concepts.

3 IE as ontology population

Historically, following the "General Theory of Terminology" created by Eugene Wüster from the late 1930s, a term is defined as a word or a group of words which corresponds to a concept in a pre-existing conceptual model. More recently, some have criticized this doctrine [32, 5]: the conceptual model and the terms are not seen anymore as absolute notions, but as the result of an artificial and application-oriented construction process based on a domain-related corpus. In other words, the terminology is not *discovered*, but *constructed*. We follow this latter conception: our conceptual model, the ontology, is seen as a specification of a normalized vocabulary, used to represent a text by neglecting some aspects of the discourse, and keeping some other ones. IE lies in the normalization of sentence fragments as ontology instances, i.e. Ontology Population.

3.1 Ontology as a normalized vocabulary

Figure 1 (full lines) exemplifies a simplified ontology of transcription in bacteria. In this model, the "transcription" of a gene (*"et"*) from a promoter (*"t_from"*) may happen

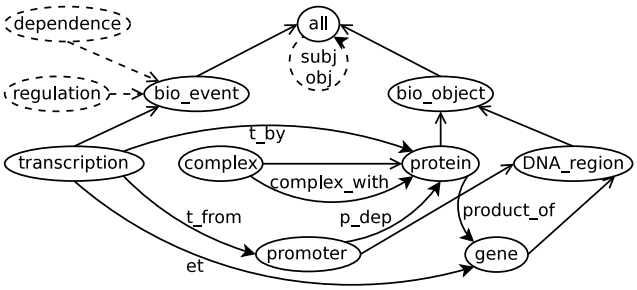

Figure 1. Sample of an ontology (full lines) and Lexical Layer (dashed lines). Labels of "is_a" relations are omitted.

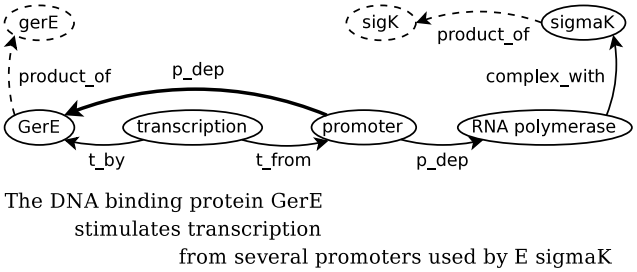

Figure 2. Example of extracted instances.

due to the action of a protein (*"t_by"*). Furthermore, a protein results from the expression of a gene (*"product_of"*), and a protein complex results from the assembly of several proteins (*"complex_with"*). These concepts and relations are seen as establishing a normalized vocabulary, used to represent the sentence shown in figure 2. Note that, as a normalized representation, not all the meaning is kept. For instance, we do not stress anymore on the "DNA binding" nature of the "GerE" protein, and the fact that the transcription of the gene occurs from "several" promoters is lost.

The OP/IE goal is to automatically produce, from Natural Language inputs, a set of instances compliant with the domain ontology. This requires complex mappings between expressions in NL to ontology structures [12], going beyond mere class/label linking (like the `rdf:label` property of RDF [6], or the more complex properties of SKOS [7]). To bridge the gap, some authors introduce lexicon models [12, 10] to ground the semantic information to the linguistic domain. We follow this approach in an IE context, by providing a Lexical Layer 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 with the ontology. 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 1). The LL is described in the same language as the ontology, so the inference rules can benefit from it.

3.2 Platform architecture

In the following, we describe an IE system architecture taking advantage of the ontology and the LL, as shown in figure 3. Provided with a NL text as input, the *ontology population module* enriches the instance base. Terms rec-

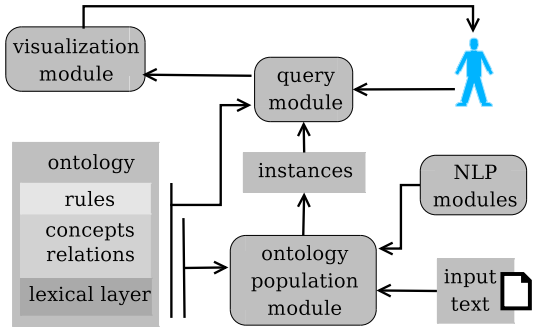

Figure 3. Ontology-based IE platform.

ognized by terminological and NER modules are normalized with the relevant instances, and properly linked with existing domain knowledge. These terms show the simplest types of linguistic variations. The OP module also completes the LL instantiation from outputs of relevant NLP modules. Then, a *query module* answers to user queries with the set of relevant instances. Those instances can result from the OP module output, from domain knowledge, or being derived by deductive inferences from the rules of the ontology. Reasoning handles more complex linguistic variations, it takes place on a semantic level, or on a syntactico-semantic level, thanks to the LL. Answers are reported to a *visualization module*, which presents them in predefined formats.

3.3 Ontology population module

The OP module links outputs from the NLP modules to the ontology. For instance, the concepts of figure 1, like the *protein* concept, will be instantiated from the outputs of a terminological module. Classes and relations of the LL are also instantiated. Dashed lines in figure 1 exemplifies the declarative definition of the LL in the ontology. The concept of "regulation" and the concept of "dependence" are specified since, see figure 4, they are required to understand the presence of an interaction between proteins GerE and EsigmaK ("stimulate" is an instance of "regulation", and "use" an instance of "dependence"). In the same way, syntactic relations between instances are needed, and specified. Domain concepts are instantiated with a terminological module, and syntactic relations with a parser. Actually, the OP module produces a set of instances as shown in figure 4.

3.4 Query module

The query module allows reasoning on the instances extracted by the OP module (figure 4). It takes advantage of the inference rules of the ontology. Consider the following user query, related to the sentence in figure 2:

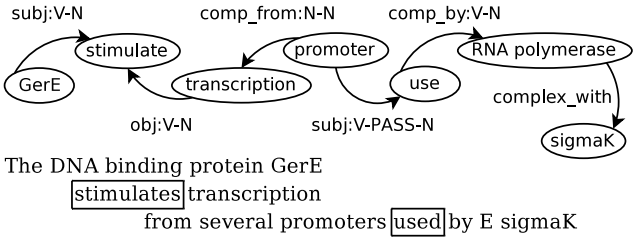

Figure 4. Ontology population module output.

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

which means "is there a protein B which binds to a promoter A?". The query module will answer:

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

The query module has infered the "p_dep" relation, in bold line in figure 2, with a rule of the ontology, such as:

$$p\_dep(B,A) \leftarrow t\_by(C,A), t\_from(C,B), protein(A), promoter(B), transcription(C).$$

It means that "if protein A is responsible for a transcription event C from promoter B, then B is dependent on (i.e. may be binded by) protein A". In the example, *p_dep(promoter, GerE)* is true as both *t_by(transcription, GerE)* and *t_from(transcription, promoter)* are true. The t_by relation was deduced from a rule like:

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

In figure 4, *t_by(transcription, GerE)* is true, as *subj(GerE, stimulate)* and *obj(transcription, stimulate)* are true. Note that this second rule uses the LL to make syntactico-semantic reasoning. Finally, the visualization module exploits the query module to produce the final output, like the graph in figure 2. In the example, class instances come from a terminological module, relations in full lines are deduced, and instances in dashed lines result from domain knowledge: the "GerE" protein is known to be encoded by the "gerE" gene, and the "RNA polymerase" protein to include the "SigK" protein — itself encoded by the "sigK" gene.

4 Application to genic interaction extraction

The modeling of genic interactions represents a considerable scientific interest for biologists, as it constitutes a fundamental step in the comprehension of the cell behaviour. The understanding of which interactions genes can establish among themselves is essential, because biological functions and pathways originate from such interaction networks.

Although, in some cases, these interactions were long studied, the main part of biological knowledge concerning them is not described in genomic databanks, but scattered throughout scientific articles. The following sentence exemplifies the complexity of the task and the benefit of an ontology-based representation:

> The degR gene is transcribed by RNA polymerase containing sigma D, and the level of its expression is low in a mecA-deficient mutant. (PMID: 10486575.)

To populate the ontology, the biological entities have to be recognized and properly linked. Simplest lexical variations (e.g. the recognition of "degR gene") are captured by NER [30]. A term–concept connection is assumed by several task-related systems, which use mere conceptual hierarchies, without relation [25, 27, 33]. In our approach, we normalize a term as a subgraph of ontology instances, including domain knowledge: in the example, the term "RNA polymerase containing sigma D" may be represented as a *protein complex* relation between an "RNA polymerase" and a "sigma D" *protein*. All synonyms have to share the same representation (e.g. "EsigmaD" or "RNA polymerase sigma D"). We emphasize the terminology status: in [27], "RNA polymerase containing sigma D" is untagged, and *protein* "sigma D" is inaccurately regarded as the interacting entity. Furthermore, whereas most terminological works focus on nouns, we handle verbal terms: the terms "transcription by EsigmaD" and "transcribed by EsigmaD" will be identically represented.

Most IE systems are limited to extract a unique relation, and face a trade-off between recall and precision. Some, for example [33], focus on precision by extracting specific interactions (e.g. a protein–protein interaction), whereas others stress on recall using general relations [27]. Here, the ontology allows to overcome this trade-off by defining various conceptual relations: a transcription event between EsigmaD and degR, and a more general regulation between the mecA mutant and the degR gene.

Finally, as we motivated it earlier, we do not only provide rules processing on a syntactico-semantic level but, using inference rules of the ontology, we can reason at a semantic level. In the example, this allows to deduce that, although the second interaction of the sentence involves an inhibition ("level of its expression is low"), as a mutant gene is implied, mecA and degR are linked by an activation. Inferences may imply multiple sentences, inducing knowledge not explicitly present in the text, as we will illustrate it in section 4.3.

Name	Example of related term
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. List of relations defined in the ontology, and phrase examples (sub-terms of the relation are shown in italic and bold).

4.1 Ontology of transcription in bacteria

We validate our system architecture by designing an ontology of gene transcription in bacteria that describes a structural model of gene and its transcription to which authors 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 do not detail the normalisation of the biological terms as subgraphs of this ontology, and we will focus on the ten defined relations in our ontology: 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), 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, a phrase example where the relation is used 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 Ontology is designed in RDF with the PROTÉGÉ editor [28]. Ternary and labeled relations are achieved by reification, and the inference rules are encoded in Prolog. The OP, query and visualisation modules are also built in Prolog. RDF to Prolog and Prolog to RDF translations are straightforwardly done, using the semantic web library of SWI-PROLOG.

4.2 Ontology learning

The inference rules of the ontology that allow to deduce these relations have to be learnt from a domain corpus. As opposed to previous approaches (see section 2), we do not use OL to acquire conceptual hierarchies or non-domain-specific relations, but learn inference rules that hold in the ontology. Learning takes place in the ontology language, and a domain expert has to provide examples defined as instantiations of the ontology. The expert creates instances of concepts and relations of the ontology from a corpus, some instances being output by the Ontology Population module. Target relations are specified to be logically implied by the inference rules; figure 5 exemplifies such annotation, the dashed lines corresponding to relations to learn.

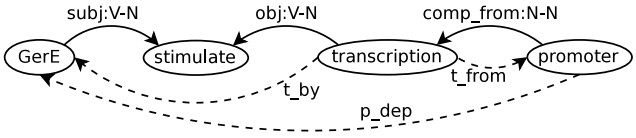

Figure 5. Learning example provided by a semantic annotation.

Learning from such a relational language is known as Inductive Logic Programming (ILP) [26], where the hypothesis and the example languages are subsets of first-order logic. Most learners handle learning in Datalog which is expressive enough for the task. In Datalog, examples are represented as closed Horn clauses, where the head of the clause is the target relation to learn. For instance, the example of the "t_by" relation in figure 5 will be equivalently represented as the following:

$$\begin{aligned} t\_by(id2, id1) \quad \leftarrow \quad & subj(id1, id3), obj(id2, id3), \\ & transcription(id2), \\ & protein(id1), \\ & regulation(id3). \end{aligned}$$

As several relations have to be learnt, learning is set into the multi-class setting where each target relation is learnt in turn, using the other ones as negative examples. 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:

$$\begin{aligned} & protein(A) \leftarrow protein\_complex(A). \\ & gene\_product(A) \leftarrow protein(A). \\ & gene\_product(A) \leftarrow rna(A). \end{aligned}$$

Processing an example involving a protein complex or a RNA, the learning algorithm now has the opportunity to choose the most relevant generality level (e.g. "protein complex", "protein" or "gene product") to learn the rules.

Relation	Recall (%)	Prec. (%)	Number
i	76.4	73.5	161
rm	90.0	90.0	17
r_dep	95.0	100.0	12
b_to	75.0	90.0	14
p_dep	91.5	94.3	47
p_of	87.5	85.2	39
s_of	61.7	80.7	21
et	95.8	99.4	168
t_from	85.0	96.7	18
t_by	65.5	82.6	44

Table 2. Results for multi-class learning. Last column shows the number of examples.

We validate the interest of using OL with a rich domain ontology by re-using the corpus of the LLL05 challenge [27]. It contains 160 sentences, in which we annotated terms, concepts and relations; 587 relations were labeled. This corpus provides dependency-like parsing of the sentences, following a normalised set of syntactic relations [4], with resolved coreferences. Output of NLP tools is complex and heavily noisy, making errors difficult to trace. Thus, to focus exclusively on the rules acquisition task, we enriched and manually curated the linguistic annotations of the LLL05 corpus.

The Lexical Layer was designed for those linguistic annotations. Briefly, we have introduced 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). A sample of the ontology instantiation allowed with this LL is given in figure 5.

Eventually, the annotated corpus was used to produce the learning set. To help learning, we added a class of non-interacting biological entities, which was generated using the closed-world assumption. We used the multi-class ILP learner PROPAL [3] to acquire a set of rules for each relation; the non-interacting class was used as negative examples each time, but was not learnt. From the 587 relations, we excluded 46 relations, matched by expert rules which exhibited recursion or dependencies with other rules. We provided PROPAL with 541 examples from ten classes, and 10155 from the non-interacting class. We used ten-fold cross-validation, averaged ten times, to evaluate recall and precision of the extraction process. The results are shown in table 2. More specific relations (et, r_dep, rm) have little lexical variability, and reach especially high scores; on the contrary, more general ones (i, t_by), exhibiting greater variability, are noticeably harder to learn. Global recall and precision, experimented by merging the ten conceptual relations into a positive label, are consistently good, as shown in table 3. Results corroborate the relevance of our OL approach. Using an ontology to specify the IE task promotes the definition of multiple specific relations, whereas related IE systems specify only one interaction relation. From a ML point of view, this approach is beneficial as it implies more general learnt rules. For instance, with a single interacting relation, the sentences "sigma(H)-dependent expression of spo0A" and "sigma(K)-dependent cwlH gene" would need two rules to be matched (basically, patterns like "A-dependent expression of B" and "A-dependent B"); however, with our ontology-defined relations, the first sentence would be matched by the rules "A-dependent B" ("i" relation) and "B of C" ("et" relation), and the second sentence by "A-dependent B" ("i" relation). Thus, in the second case, the "i" rule matches two sentences, where two rules were previously needed.

Recall (%)	Prec. (%)
89.3	89.6

Table 3. Results for two-class learning.

4.3 Visualizing extracted instances

Using reasoning capabilities of an ontology allows a richer usage of the extracted knowledge. From the four following sentences:

> (1) SpoIIID binds strongly to two sites in the cotC promoter region; (2) SpoIIID represses cotC transcription by sigma(K) RNA polymerase; (3) Transcription of cotC by sigmaK RNA polymerase is activated by GerE; (4) GerE represses transcription from the sigK promoter;

the system extracts the genic interacting couples *(spoIIID,cotC)*, *(gerE,cotC)*, *(gerE,sigK)* and *(sigK,cotC)*. Still, thanks to the ontology-based representation, the output of the system can be far more expressive. Figure 6 illustrates some outputs of the visualization module, following an increasing refinement degree. Graph 6(a) only shows the interaction between the spoIIID and cotC genes, whereas 6(b) figures a more precise description. spoIIID and cotC interact as: (i) the SpoIIID protein is the product of the spoIIID gene (relation product_of, from domain knowledge); (ii) a promoter depends on spoIIID (p_dep, inferred and not explictly present in the text); (iii) the promoter is part of the cotC gene (p_of, extracted from sentence 1). The representation of 6(c) is even more specific: the promoter of cotC depends on SpoIIID as the protein binds to (b_to) a site included (s_of) in the promoter. These two relations were extracted from sentence 1, with the following learned rules:

$$b\_to(A,B) \leftarrow subj(C,A,'V-N'), \\ comp(C,B,'V-N',to)$$

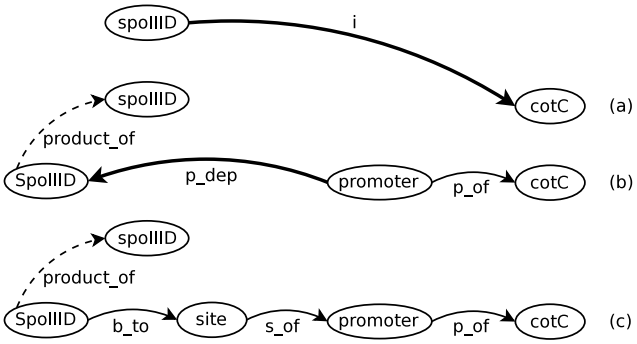

Figure 6. Visualization examples.

$$s\_of(A,B) \leftarrow comp(A,B,'N-N',in), concept(A,dna\_reg)$$

Actually, the visualization module produces genic networks like shown in figure 7. Note that inferences are not restricted to a sentence: for instance, as the sentence 3 asserts that cotC transcription is activated by GerE, it is possible to deduce that it happens from the cotC promoter (t_from). This latter deduction permits to conclude that the cotC promoter is dependent (p_dep) of GerE. Implicit knowledge distributed into two sentences is therefore made explicit.

5 Conclusion and perspectives

Emerging IE applications, like ones from the biomedical domain, require complex knowledge representations, to both efficiently extract information from text and present exploitable information to the user. Ontology is a well-motivated formalism to represent such complex knowledge, but its tight integration to IE systems is still a research issue. In this paper, we proposed an original integration where an IE system is fully specified by designing an ontology, and its conception is an Ontology Population task. In this context, the inference rules of the ontology subsume classical extraction rules or patterns by providing access to appropriate abstraction level, domain knowledge and semantic inference capabilities of ontology knowledge representation languages. Inference rules reasoning on the ontology can gather information scattered throughout multiple sentences, hence discovering knowledge not explicitly stated in texts. We demonstrate the benefit of our approach by extracting genic interactions from an annotated corpus related to *Bacillus subtilis*. Our system provides representation and inference capabilities beyond current IE systems, at least in the biological domain: it achieves a good trade-off between recall and precision by using multiple domain-specific relations to model the complexity of transcription in bacteria; its output may be represented according to multiple abstraction levels, giving detailed explanations of the extraction.

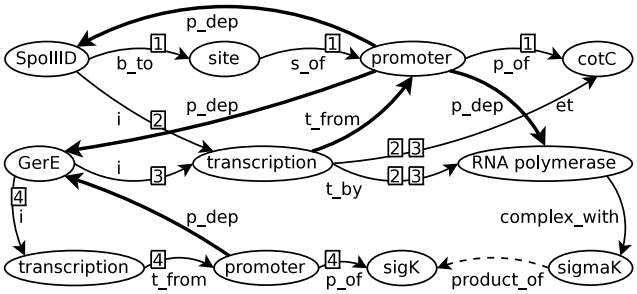

Figure 7. Extracted genic interaction network.

In the future, the declarative nature of our system will allow its easy extension to other tasks and domains. Related to genic interactions, we want to handle more phenomenons, especially inhibition/activation distinction, and non-genic actors (e.g. environmental factors). From an operational perspective, we aim at fully automatizing the OP module by linking it to available NLP pipelines.

Finally, we used a relational learning system to learn the inference rules of the ontology. A benefit of our approach is that it casts the important work on extraction pattern learning into the Ontology Learning paradigm. On one hand, as noted by [8], OL can benefit from "evaluation methods that are central to most machine learning work", but on the other hand Machine Learning can benefit from Ontological Engineering in the development and motivation of knowledge representation languages. Notably, as the representation choice is a crucial step in ML, its declarative definition through the ontology is a significant contribution to the domain. Therefore, we plan more works on the text representation for learning inference rules, through a comparative study of several Lexical Layers.

Acknowledgements

We thanks INRA to have awarded a Doctoral and Postdoctoral Fellowship to Alain-Pierre Manine.

References

[1] H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[2] E. Alphonse, S. Aubin, P. Bessières, G. Bisson, T. Hamon, S. Lagarrigue, A. Nazarenko, A.-P. Manine, C. Nédellec, M. O. A. Vetah, T. Poibeau, and D. Weissenbacher. Event-based information extraction for the biomedical domain: the Caderige project. In *Proc. Intl. Workshop NLP in Biomedicine and its Applications*, pages 43–49, 2004.

[3] E. Alphonse and C. Rouveirol. Extension of the top-down data-driven strategy to ILP. In *Proc. Conf. Inductive Logic Programming*, pages 49–63, 2006. Springer Verlag.

[4] S. Aubin. LLL challenge — Syntactic analysis guidelines. Technical report, LIPN, Univ. Paris 13, 2005.

[5] D. Bourigault and C. Jacquemin. Construction de ressources terminologiques. In J.-M. Pierrel, editor, *Ingénierie des langues*, pages 215–233. Hermès Science, 2000.

[6] D. Brickley and R. Guha. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, 2004.

[7] D. Brickley and A. Miles. SKOS core vocabulary specification. W3C working draft, 2005.

[8] P. Buitelaar, P. Cimiano, and B. Magnini. Ontology learning from text: An overview. *Ontology Learning from Text: Methods, Evaluation and Applications*, vol. 123 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2005.

[9] P. Buitelaar, P. Cimiano, S. Racioppa, and M. Siegel. Ontology-based information extraction with SOBA. In *Proc. Intl. Conf. Language Resources and Evaluation*, pages 2321–2324, 2006.

[10] P. Buitelaar, M. Sintek, and M. Kiesel. A multilingual/multimedia lexicon model for ontologies. *Proc. 3th Eur. Semantic Web Conference*, volume 4011 of *LNCS*, pages 502–513. Springer Verlag, 2006.

[11] P. Cimiano. Ontology-driven discourse analysis in GenIE. In A. Düsterhöft and B. Thalheim, editors, *Proc. 8th Intl. Conf. Applications of Natural Language to Information Systems*, volume 29 of *LNI*, pages 77–90. GI, 2003.

[12] P. Cimiano, P. Haase, M. Herold, M. Mantel, and P. Buitelaar. LexOnto: A model for ontology lexicons for ontology-based NLP. In *Proc. OntoLex07 Workshop*, 2007.

[13] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo. Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics*, 20(5):604–611, 2004.

[14] C. Friedman, P. Kra, and A. Rzhetsky. Two biomedical sublanguages: a description based on the theories of Zellig Harris. *J. Biomedical Informatics*, 35(4):222–235, 2002.

[15] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(Suppl. 1):S74–S82, 2001.

[16] R. Gaizauskas, G. Demetriou, P. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The PASTA system. *Bioinformatics*, 19(1):135–143, 2003.

[17] A. Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, 1999.

[18] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *Proc. Eur. Conf. Knowledge Acquisition and Management*, 2002.

[19] S. Huffman. *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, chapter Learning Information Extraction Patterns from Examples. Springer Verlag, 1996.

[20] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, 1995.

[21] Y. Li, K. Bontcheva, and H. Cunningham. Hierarchical, Perceptron-like Learning for Ontology Based Information Extraction. In *16th Intl. World Wide Web Conf. (WWW2007)*, pages 777–786, 2007.

[22] D. Lin and P. Pantel. DIRT – discovery of inference rules from text. In *Proc. 7th Intl. Conf. Knowledge Discovery and Data mining*, pages 323–328, 2001. ACM.

[23] A. Maedche, G. Neumann, and S. Staab. Bootstrapping an ontology-based information extraction system. In *Intelligent exploration of the web*, pages 345–359, 2003.

[24] D. McGuinness and F. van Harmelen. OWL web ontology language overview: W3C recommendation 10 february 2004. Technical report, W3C, February 2004.

[25] Y. Miyao, T. Ohta, K. Masuda, Y. Tsuruoka, K. Yoshida, T. Ninomiya, and J. Tsujii. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In *Proc. COLING-ACL 2006*, pages 1017–1024, 2006.

[26] S. Muggleton and L. D. Raedt. Inductive Logic Programming: Theory and methods. *J. Logic Programming*, 19,20:629–679, 1994.

[27] C. Nédellec. Learning language in logic — Genic interaction extraction challenge. In J. Cussens and C. Nédellec, editors, *Proc. 4th Learning Language in Logic Workshop (LLL05)*, pages 31–37, 2005.

[28] N. Noy, M. Sintek, S. Decker, M. Crubézy, R. Fergerson, and M. Musen. Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[29] K. Oda, J.-D. Kim, T. Ohta, D. Okanohara, T. Matsuzaki, Y. Tateisi, and J. Tsujii. New challenges for text mining: mapping between text and manually curated pathways. *BMC Bioinformatics*, 9(Suppl 3):S5, 2008.

[30] J. Park and J.-J. Kim. *Text Mining for Biology*, chapter Named Entity Recognition. Artech House Books, 2006.

[31] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. KIM: a semantic platform for information extraction and retrieval. *Natural Language Engineering*, 10(3-4):375–392, 2004.

[32] F. Rastier. Le terme: entre ontologie et linguistique. In *La banque des mots*, pages 35–65. CILF, 1995.

[33] J. Saric, L. Jensen, R. Ouzounova, I. Rojas, and P. Bork. Large-scale extraction of protein/gene relations for model organisms. In *1st Intl. Symp. Semantic Mining in Biomedicine*, 2005.

[34] J. Völker, D. Vrandecic, Y. Sure, and A. Hotho. Learning disjointness. In E. Franconi, M. Kifer, and W. May, editors, *Proc. 4th Eur. Semantic Web Conf.*, volume 4519 of *LNCS*, pages 175–189. Springer Verlag, 2007.