Ontology-based information extraction of regulatory networks from scientific articles with case studies for Escherichia coli

Antonio Moreno, David Isern, Alejandra C. López Fuentes

Universitat Rovira i Virgili, Department of Computer Science and Mathematics, ITAKA research group, Avda. Països Catalans, 26, 43007 Tarragona, Catalonia, Spain
Programa de Genómica Computacional, Centro de Ciencias Genómicas, Universidad Nacional, Autónoma de México, Cuernavaca, Morelos, México

Abstract

The amount of scientific papers in the Molecular Biology field has experienced an enormous growth in the last years, prompting the need of developing automatic Information Extraction (IE) systems. This work is a first step towards the ontology-based domain-independent generalization of a system that identifies Escherichia coli regulatory networks. First, a domain ontology based on the RegulonDB database was designed and populated. After that, the steps of the existing IE system were generalized to use the knowledge contained in the ontology, so that it could be potentially applied to other domains. The resulting system has been tested both with abstract and full articles that describe regulatory interactions for E. coli, obtaining satisfactory results.

1. Introduction

The Web is a vital tool in our Information Society. It has grown exponentially through the 1990s and into the 21st century and it has expanded to serve hundreds of millions of users and a multitude of purposes in all parts of the world. The Web carries a vast range of information resources that provide general information which can be easily found on any subject with the simple click of the mouse. Thousands of new Web pages appear every day: news, magazines, and journal articles are constantly being created online. All these trends result in an enormous amount of text available in digital form; however, these repositories of text are mostly untapped resources of information, and identifying specific desired information in them becomes a difficult task.

The Web is drowning us with an enormous amount of pages of information related to all domains of discourse. Recently, the fields of Molecular Biology and Medicine have enjoyed an explosive development; as a result, the number of biomedical journal articles has grown exponentially, and an urgent need on the part of biologists to find specific information in the ever expanding biological literature has emerged.

Access to full-text articles is difficult because each journal has its own organization, interface and formatting conventions. Fortunately, in the Biology and Medicine fields the abstracts of the scientific papers are collected and indexed in MEDLINE. This repository, hosted at the National Library of Medicine (NLM), is the primary component of PubMed that indexes over 5000 different journals in Medicine and Molecular Biology. It currently contains more than 22 million abstracts and it keeps growing.

Today, it is possible to retrieve articles using queries that can be very complex (including different attributes of the publication, combined with logical operators) and to read their abstracts. With the increasing distribution of journals in electronic format, the full text of any paper will soon be only one click away (Miguel A. Andrade & Bork, 2000). However, the easy access to the literature does not solve the problem of the selection of information. Reading or even browsing all of those publications is something that most researchers will not contemplate. Clearly, there is a need of developing methods for automatic extraction of relevant information from any source of scientific data, especially sources such as scientific literature written in natural language.

Molecular Biology is a branch of Biology which primarily deals with functions, characteristics and structures of mainly three major macro-molecules: DNA, RNA and proteins. Molecular research technologies and developments in Information Technologies have combined to produce a huge amount of information related to the multiple research areas of Molecular Biology such as sequence analysis, genome annotation, analysis of gene expression, analysis of regulation, etc. Several databases have been designed and implemented in order to compile all this information (Benson, Karsch-Mizrachi, Lipman, Ostell, & Wheeler, 2008; Grivell, 2002; Keseler et al., 2005). However, the amount of knowledge in the domain is overwhelming and grows at an unprecedented rate. Much of it...
is published and available electronically in the form of scientific
texts, therefore, in natural language.

The main hypothesis in this work is that it is necessary to devel-
oped improved computer-based tools to aid human experts to extract
the facts needed by the research community (Collado-Vides et al.,
2009). In the recent years, Information Extraction (IE) has emerged
within the Artificial Intelligence community to face this problem.
IE systems usually follow a sequence of steps that gradually iden-
tify pieces of information, analyze them, and finally represent them
in a structured way. This paper presents an evolution of those tradi-
tional systems. It shows the current implementation of an IE sys-
tem to extract regulations on the biological domain using some
language-dependent elements, CCG-IE, developed in the UNAM’s
Centre for Genomic Sciences, and proposes an evolution to add a
higher level of expressivity using a widely-used model of knowl-
edge representation (ontologies). The main goal of the work has
been to create an ontological structure used to guide the extraction
of information. The design and use of this ontology is crucial to cre-
ate a more powerful system able to extract information from dif-
ferent domains.

The paper explains in detail the design and the main classes of
the domain ontology. In this case, due to the complexity of the do-
main of research, the ontology has been constructed from a large
database of gene regulations (RegulonDB). Additionally to this cor-
pus of terms, the ontology requires more elements such as the rep-
resentation of grammars used to annotate sentences in natural
language-based texts. The paper also describes the ontology-based
IE system and the tests that have been performed on the current
prototype.

The rest of the paper is organized as follows. First of all, Section 2
presents the state of the art on ontology-based IE, paying special
attention to the available IE techniques and their application in
Molecular Biology. Section 3 describes the domain ontology, which
is used later to guide the IE process (Section 4). Section 5 presents
some case studies of the current prototype and, finally, Section 6
lists some conclusions and lines of future research.

2. Related work

This section provides an introduction to Information Extraction
a classification of the main techniques used in this field (statistical,
rule-based and ontology-based) and a general architecture of
ontology-based IE systems (Wimalasuriya & Dou, 2010). Then, an
outline of five recent biomedical Information Extraction systems
is performed, and a comparative table of the analyzed systems
with the one proposed in this paper is given.

2.1. Information extraction techniques

Information Extraction is concerned with selecting salient facts
about a given topic from a set of documents. Typically, these facts
are then entered automatically into a database, which may then be
used for further processing. IE is a technology based on natural lan-
guage analysis. An IE system must be designed to extract the enti-
ties and relations appropriate to a specific task (Blaschke,
Hirschman, & Valencia, 2002). In general, the core of an IE system
is an extractor which processes text; it overlooks irrelevant words
and phrases and attempts to home in on entities and the relation-
ships between them (Etzioni, Banko, Soderland, & Weld, 2008).
There are several methods to perform IE which can be classified
into statistical methods and rule-based methods, being the later
superior to the former in those tasks that require the extraction of
relationships (expression of a specific product of a certain gene
in a certain tissue, specific function of a protein, etc.), although
they are harder to implement (Feldman, Regev, Finkelstein-Landau,
Hurvitz, & Kogan, 2002).

Additionally, another approach has recently emerged: ontol-
ogy-based information extraction (OBIE). Since IE is essentially
concerned with the task of retrieving information for a particular
domain, specifying the concepts of that domain formally and
explicitly through an ontology can be helpful to this process (Wim-
 alasuriya & Dou, 2010).

2.1.1. Statistical information extraction

The statistical approach uses training material which can be
annotated (usually through Machine Learning techniques). Several
methods are based on the frequency of co-occurrence of words in a
large text corpus. The basic elements of text are words, and their
frequencies, co-occurrences and lexical features can be used to
cluster and classify text, find documents that treat a similar theme
or select words that describe a group of documents. For example,
M.A. Andrade and Valencia (1998) used the characteristics of word
distributions in text clusters to extract significant words. These ap-
proaches are limited because words are often ambiguous and refer
to more than one object. Moreover, different words can have the
same meaning (synonyms) and the same word can be part of con-
structions with very different meanings.

There are many statistical methods that have been applied for
specific tasks in the IE process and have proved to be very effective
where there is a large corpora of training data available (Bikel,
Mill-
apply statistical methods to automatic speech recognition. They use
smoothed n-grams in order to find the most probable string of
words from a set of candidate strings. Merialdo (1994), as part of
speech tagging, uses hidden Markov models to find the most prob-
able sequence of tags in a given sequence of words. In a more com-
plex domain, syntactic parsing using probabilistic grammars
permits to find the most probable parse tree given a word sequence
(Charniak, 1997). These methods have also been used for word dis-
ambiguation (Buscaldi, 2009; Stokoe, Oakes, & Tait, 2003).

2.1.2. Rule-based information extraction

Rule-based Information Extraction is a process by which struc-
tured entities are extracted from text based on rules written by hu-
mans developers that capture syntactical, lexical and semantic
knowledge required to identify the entities and the relationships
in the domain (Feldman et al., 2002). The compositional nature
of rule-based information extraction also allows rules to be ex-
pressed over previously extracted entities (Michelakis, Krishna-
murthy, Haas, & Vaithyanathan, 2009).

Fig. 1 shows the schematic diagram of Rule-based IE, which has
five processes and uses dictionaries or lexicons that are created by
experts of a specific domain.

(a) Tokenization: Identifying the boundaries of sentences in a
document and decomposing each sentence into tokens. Tokens are
obtained by splitting a sentence along a predefined set of delimiters
like spaces, commas, and dots. A token is typically a word, a digit, or
punctuation.

(b) Part-of-speech (POS) tagging: assigning to each word a
grammatical category coming from a fixed set. The set of
tags includes the conventional parts of speech such as noun,
verb, adjective, adverb, article, conjunct and pronoun.

(c) Semantic tagging: recognizing entities relevant to the
domain. For this step it is common to use dictionaries or
gazetteers that contain relevant terms and are created by
experts of a specific domain.

(d) Parsing: grouping words in a sentence into prominent
phrase types such as noun phrases, prepositional phrases,
and verb phrases. A context-free grammar is typically used

to identify the structure of a sentence in terms of its constituent phrase types. The output of parsing is a parse tree that groups words into syntactic phrases.

(e) Relationship extraction: the resulting parse trees from the previous step are useful in entity extraction because typically named entities are noun phrases. In relationship extraction they are useful because they provide valuable linkages between verbs and their arguments.

2.1.3. Ontology-based information extraction

The detection and extraction of relevant information from textual documents depends on the proper understanding of text resources (Sánchez, Moreno, & Del Vasto Terrientes, 2012). Rule-based IE systems are limited by the rigidity and ad-hoc nature of the manually composed extraction rules. As a result, they present a very limited semantic background.

Ontology-based information extraction (OBIE) has recently emerged as a subfield of IE. Here, ontologies are used by the information extraction process and the output is generally presented through an ontology. Studer, Benjamins, and Fensel (1998) describe an ontology as “a formal, explicit specification of a shared conceptualization”. Conceptualization refers to an abstract model which makes it possible for concepts to be defined as well as described. Complex concepts can be built on top of simpler concepts. Furthermore, reasoners can be used to check whether or not all the statements and definitions in the ontology are mutually consistent.

2.1.3.1. Characteristics of ontology-based IE systems. The key characteristics of an OBIE system, as proposed by Wimalasuriya and Dou (2010), are the following:

- Process unstructured or semi-structured natural language text.
- Present the output using ontologies.
- Use the ontology to guide the IE process.

A related issue is the use of the term ‘ontology-driven information extraction’ (McDowell & Cafarella, 2006; Wu, Hoffmann, & Weld, 2008; Yildiz & Miksch, 2007a). In most cases, this can be seen as a synonym for OBIE, which has emerged due to the lack of a standard terminology. In this work, the term ontology-based information extraction is used since it appears to be the one used by a majority of publications. However, Yildiz and Miksch (2007b) make a distinction between these two terms. They state that in ontology-driven systems the extraction process is driven by an ontology whereas the ontology is yet another component in an ontology-based system.

Combining these factors with the definitions of information extraction presented by Russell and Norvig (2003) and Riloff (1999), an OBIE system can be defined as process that analyzes unstructured or semi structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies (Wimalasuriya & Dou, 2010).

2.1.3.2. Common architecture of OBIE systems. Fig. 2 represents the union of different components found in different OBIE systems. As such, many systems do not contain all the components of this architecture, e.g. the systems that use an ontology defined by others, instead of constructing an ontology internally, do not have the ‘ontology generator’ component.

As represented in the figure, the textual input of an OBIE system first goes through a preprocessor component, which converts the text to a format that can be handled by the IE module, e.g. converting a PDF file into a pure text file.

The information extraction module is where the actual ontology-based extraction takes place. The ontology that is used by the system may be generated internally by an ontology generator component. This process might also make use of a semantic lexicon. In addition, humans may assist the system in the ontology generation process. This is typically done through an ontology editor such as Protégé2. Humans may also be involved in the

\[\text{Fig. 1. Rule-based IE stages.}\]

\[\text{Website: http://protege.stanford.edu (Last access: October 22nd, 2012).}\]
information extraction process in some systems that operate in a semiautomatic way. The output of the OBIE system consists of the information extracted from the text. It can be represented in an ontological structure, usually stored in some kind of knowledge base.

2.2. Information extraction in molecular biology

IE has recently been applied in Bioinformatics. In particular, there have been several works that aim to extract information from scientific papers on Molecular Biology. The most relevant ones are summarized in this section.

2.2.1. GenWays

GENWAYS (Genomic pathways, Rzhetsky et al. (2004)) is a fully automated system that extracts and structures information related to molecular pathways. The system has an agent that crawls different knowledge sources. The first stage of the process is to identify biological terms inside those documents. A synonym/homonym module clarifies the meaning by assigning a "canonical" name to the multiple aliases of each concept. Then, a term classifier module uses the context to resolve sense ambiguity. The next step tries to identify relationships between all identified elements using GENIES (Friedman, Kra, Yu, Krauthammer, & Rzhetsky, 2001). GENIES is a natural-language processing parser that takes as input plain text with identified and tagged concepts, and its output is a tree-based structure that represents complex nested relationships.

This output may contain incorrectly extracted statements. A curation module automatically annotates the original statements with statements regarding the confidence in the corresponding information. In this way, the user receives only the most confident suggestions.

2.2.2. BioRAT

Biological Research Assistant for Text mining (Corney, Buxton, Langdon, & Jones, 2004) is a tool designed to perform biomedical IE, which is able to locate and analyze both abstracts and full-length papers. The heart of BioRAT is an IE engine based on GATE (General Architecture for Text Engineering, (Cunningham, Maynard, Bontcheva, & Tablan, 2002)), which is a general purpose text engineering system. BioRAT uses GATE to label words according to their parts of speech, and then applies a filter that rejects determinants, verbs, etc. as not being proteins. It has a Molecular Biology general orientation and a friendly interface. The components of GATE that BioRAT modifies are gazetteers and templates. A gazetteer is a list of words identifying members of a particular category. BioRAT incorporates gazetteers from three sources, namely MeSH, Swiss-Prot and hand-made lists.

A template is a representation of a text pattern that allows extracting information automatically. It consists of a number of predefined slots to be filled by the system from information contained in the text, e.g. ‘interaction of’ (PROTEIN_1) ‘and’ (PROTEIN_2). Here, ’PROTEIN_1’ and ’PROTEIN_2’ are slots to be filled with names of proteins, as defined by a gazetteer. Templates are written by hand; therefore, BioRAT incorporates a template design tool with a graphical user interface.

BioRAT produces data in XML format, which can be imported into existing database systems. The same data are produced simultaneously as HTML for viewing in applications such as a browser, if that is more convenient for the user. Each record in the resulting database represents a single completed template.

2.2.3. BioText

BioText combines structural natural language processing with Machine Learning methods to address the general and domain-specific challenges of Information Extraction targeting protein–protein interactions (Ginter et al., 2005). As depicted in Fig. 3, BioText includes different techniques such as information retrieval, named entity recognition, syntactic analysis, and pattern-based domain analysis.

The system works with two inputs: a set of annotated resources and a list of documents. The manually-annotated corpus is a domain-language set of biomedical resources. It is focused on protein–protein interactions and contains named entities, dependencies and entity interactions (Pyysalo et al., 2007).

The information retrieval part detects sentences that contain domain-related pieces of information by applying specific rules. BioText uses a rough-set classifier that receives names of proteins, verbs related to protein–protein interactions, and their mutual
positions. The next stage is the recognition and disambiguation of named entities (NEs). The determination of the boundaries of the NEs and their classification into classes as genes and proteins is an arduous task due to the vagueness or inexistence of conventions in this domain. BioText proposes a kernel-based learning algorithm to address the disambiguation problem. The set of NEs retrieved from the previous stage are used as input of the syntactic analysis. BioText uses the English-centred link grammar parser (Grinberg, Lafferty, & Sleator, 1995) to augment input sentences with tokenization processing data. Finally, BioText uses a set of handwritten patterns to extract factual knowledge from the parsed sentences. Each pattern declares a substructure of the linkage (a link grammar-based representation) that is likely to state a protein–protein interaction.

2.2.4. String-IE

String-IE (Šarić, Jensen, Ouzounova, Rojas, & Bork, 2006; Šarić, Jensen, & Rojas, 2005) is a rule-based system which implements cascaded finite state automata (Abney, 1996) to extract regulatory gene/proteins networks from biomedical abstracts and papers.

As shown in Fig. 4, String-IE is organized in cascade modules. The set of terms of interest are then annotated with part-of-speech and lemma information, using the biomedical semantic corpus Genia 3.0. At this point, the system is able to detect gene/protein names. This stage uses a list of synonyms manually composed with orthographic variants of names. In addition to the recognition of the gene and protein names, the semantic tagging module recognizes several other terms (e.g. nouns representing highly relevant concepts, nouns triggering experimental or artificial contexts, enzyme names) and annotates them with semantic tags. (See Fig. 5).

The next stage consists in the extraction of named entities, which chunks noun phrases generalizing over both POS-tags and semantic tags. This syntactic–semantic chunking is performed to recognize named entities using cascades of finite state rules implemented as a CASS grammar (Abney, 1996). Then, the protein–gene focused detection of relations between entities is performed. This processing step detects three types of relations between the recognized named entities: up-regulation, down-regulation, and (unspecified) regulation of expression. Syntactic and semantic properties of the relevant verbs are combined to map each of them to one of the three relation types. At the end, the user receives a structured and relation-annotated text.

Fig. 3. General architecture of BioText (from Ginter et al. (2005)).

Fig. 4. General architecture of String-IE (Šarić et al., 2006).
2.2.5. Gene regulatory network system

The gene regulatory network system (GRNS) (Song & Chen, 2009) consists of four modules: the knowledge collection and creation, pre-processing, information extraction, and post-processing. The external database level includes some general database resources, such as GeneBank and Swiss-Prot, as well as some specific organism resources.

The IE Module does the tokenization, sentence splitting and part-of-speech tagging. It also recognizes the gene and the protein entities, and extracts the relations and other kinds of entities based on a rule-based approach. For the POS tagging it uses the Brill POS tagger (Brill, 1995); after POS tagging, the IE module performs term recognition and variant detection to recognize gene/protein names.

To deal with problems such as abbreviations, homology and aliases, a gene dictionary of aliases and abbreviations is used. This dictionary was constructed by combining multi-database resources, such as the Swiss-Prot and the *Pseudomonas aeruginosa* Genome Database.

GRNS used the cascaded finite state automata implemented by a CASS parser (Abney, 1996) to recognize the gene regulatory relation and phenotype information. Other systems such as the String-IE also use the CASS parser to recognize the regulatory gene/protein relation. After the IE process, GNRS automatically constructs and visualizes, in a graphical form, the regulatory networks based on the entity and relation information extracted from biomedical literature.

![Fig. 5. General architecture of GNRS (Song and Chen, 2009).](image-url)
2.3. Comparison and discussion

Although the systems described above have been constructed in different ways, they all fit in a generic IE architecture. Table 1 summarizes their main features and compares them with the proposal of this paper. String-IE and GRNS tightly follow the ruled-based architecture, and each of their modules represents a step in the architecture (tokenization, POS tagging, semantic tagging,
parsing and relation extraction). BioRAT also realizes these steps through the GATE engine, besides having a template design module. On the other hand, GenWays encapsulates tokenization and POS tagging in its preprocessor module, which in turn uses the term tagger module; the rest of the steps are included in the parser module. BioText carries out the IE task in a different way and order that combines rule-based IE with Machine Learning methods. As shown at the bottom of the table, our proposal makes a step forward with respect to the previous proposed works in two fields: use a domain ontology as knowledge representation structure, and export the results in a structured-based language like XML, which can be employed by other systems.

3. Generation of an ontology from RegulonDB

The ontology used to guide the extraction of information has three main parts (see Fig. 6). The first one includes the concepts and relationships extracted from RegulonDB concerning Molecular Biology. The second part, named lexicon, stores terms and tags from the dictionaries used by the CCG-IE system and thus makes the system domain independent. The third part, named grammar, includes a set of general rules used to recognize biological concepts. The next section introduces the basic content of RegulonDB, whereas the following sections describe the three components of the ontology.

3.1. RegulonDB

RegulonDB\(^3\) (Gama-Castro et al., 2011) is an open database that offers information about the elements and interactions of the network of transcriptional regulation in Escherichia coli K-12. The information in RegulonDB is structured in an entity-relationship model that is composed by about thirty one tables.

Fig. 7 shows the portion of the model which corresponds to the tables that store most of the information about gene regulation. The tables in orange are the ones used to generate the ontology to be used by the IE system.

The Gene, Terminator, Attenuator, Attenuator Terminal, Riboswitch, Shine Dalgarno, Product, Motif, Transcription Factor, Effector, Promoter and Site tables contain single elements of biological entities, whereas the Transcription Unit and Conformation tables represent sets of entities with biological meaning.

The following are the biological definitions of the orange-colored tables:

- **Gene**: Segment of DNA involved in producing a polypeptide chain or stable RNA. It includes the regions preceding and following the coding region (leader and trailer).
- **Promoter**: DNA sequence where RNA polymerase binds and initiates the transcription of a particular gene. Promoters are located near the genes they regulate.
- **Site**: The binding sites are physical DNA sites recognized by transcription factors within a genome.
- **Transcription Unit (TU)**: Set of one or more genes transcribed from a single promoter. A TU may also include regulatory protein binding sites affecting this promoter and a terminator.
- **Product**: RNA or protein based on the gene template.
- **Transcription factor (TF)**: Complex protein that activates or represses the transcription of a TU upon binding to specific DNA sites.
- **Effector**: Molecule that binds to a protein (forming a conformation) and thereby alters the activity of that protein.
- **Conformation**: Physical object whose structure is comprised of other physical objects bound to each other non-covalently, at least one of which is a macromolecule (e.g. a protein).
- **Regulatory Interaction**: Regulation of the expression reaction by the controlling element such as a transcription factor.

Additionally, an object synonym table stores a list of synonyms of genes, products and transcription units. Synonyms are additional names given to these biological entities in other databases. For instance, the gene name adhE has the synonyms adhC, ana and b1241. The next section describes the creation of the ontology that will guide the IE process.

3.2. Translating RegulonDB into an ontology

The translation of some of the RegulonDB tables into an ontology is an automated process implemented in Java\(^4\). First, a set of rules defines the correspondence between the tables in the database and the ontology classes. After creating the classes, some properties and relationships are added to the ontology. Finally, a crawler permits to collect data from the database and populate the ontology.

Table 2 summarizes the criteria used to create the ontology-based representation of the RegulonDB core.

Some numerical data about the instances (individuals) generated in this part of the ontology: 4611 Genes, 177 Transcription Factors, 3409 Transcription Units, 4309 Proteins, 167 RNAs, 1940 Binding Sites, 76 Effectors, 232 Conformations, and 1878 Promoters.

3.3. Lexicon class

The PhysicalObject class described in the previous section contains all the biological entities that should be recognized in the scientific articles. Lexicon is a class placed at the same level (see Fig. 6). It was created to store terms and tags from the dictionaries used by the CCG-IE system, in order to make the system domain independent. Its three subclasses (MultiUnit, Biological and Parse-Terms) correspond to three dictionaries of the CCG-IE system.

- **MultiUnit**: This class contains multi-word biological terms (e.g. “Transcriptional activation”). It was automatically populated with a gazetteer containing 4053 terms (one instance per term).
- **Biological**: It contains biological terms that are relevant for the domain, although they do not correspond to physical objects (e.g. regulons, phosphorylated). A file containing 57497 words, which has been manually depurated, permits to create 31689 ontology instances.
- **ParseTerms**: This class stores tags needed by the parser to map from a standard tagset to a grammar-specific tagset, as will be explained in the next section.

3.4. Grammar class

This class is also located in the same level than Lexicon and PhysicalObject (see Fig. 6). It contains grammar rules focused on the expression of biological concepts. These rules, as will be explained in the next section, are classified into a sequence of levels since it is a requirement of the parsing program. In the ontology each level is represented as a subclass of the Grammar class, giving a total of 46 subclasses (46 levels) with their instances which are the grammar rules.

\(^3\) Webster: http://www.regulondb.ccg.unam.mx/ (Last access, October 23rd, 2012).

\(^4\) The whole ontology as well as some screenshots of the ontology can be found at http://tinyurl.com/ITAKAre gulonDB (Last access: October 23rd, 2012).
4. Improving and generalizing the CCG-IE system

As mentioned in the introduction, UNAM’s Centre for Genomic Sciences (CCG) has started to develop processes that allow the automation of the curation process, so that information about the interactions of the regulatory network can be automatically extracted by analyzing scientific articles. The main goals of this work are to improve these processes by following an OBIE approach, and to generalize the system in order to make it applicable to other domains of knowledge.

4.1. CCG-IE system architecture

The CCG-IE system (Rodríguez-Penagos, Salgado, Martínez-Flores, & Collado-Vides, 2007a,b) was implemented using the rule-based approach, because it was more focused on accuracy (precision) than on coverage (recall), and rule-based systems are more accurate for well-defined tasks (although they are very labor-intensive to implement). Statistically-based approaches, on the other hand, despite being less precise, are more robust and tolerant to noisy data and errors.

The CCG-IE system creates computer-readable networks of regulatory interactions directly from different collections of abstracts and full-text papers. It was developed by customizing and extending the String-IE (see Section 2.2.4) system for the analysis of E. coli K-12 literature. The basic processing pipeline is shown in Fig. 8.

The system input consists of a set of articles in plain text. The phases of the system are the acquisition of the input corpus, its preprocessing, the analysis of lexical components (POS tagging), a primary semantic analysis (semantic tagging, linking semantic tags to the components of each sentence), full syntactic analysis (parsing) and a final step which extracts the regulatory network interactions. All these steps are commented in the following sections, which describe the initial design of each step and how they have been improved and generalized so that the system is applicable to other domains.

4.2. Preprocess

The aim of this initial module is to normalize and tokenize the input plain text file. Some examples of normalization are removing punctuation, removing accent marks and other diacritics from letters, identifying and expanding contractions, etc. The tokenization
separates all words and terms, dealing with abbreviations and punctuation and identifying sentential boundaries.

The second preprocessing step is more domain-oriented and consists of recognizing specific biological multi-unit terms, this is, terms that have more than one word (e.g. “DNA binding domain”, “negative regulation”, “inner membrane protein”) and place them in the same line of the file for tagging purposes.

Originally, the CCG-IE system used a dictionary with all the existing multi-unit terms to recognize them in the text file. Since one of the purposes of this work is to generalize the system and implement an ontology-based approach, this dictionary was mapped into an ontology class (MultiUnit, see Section 3.3) and automatically populated with its corresponding instances (each multi-unit term).

After executing these two tasks, the final output is a tokenized file (.tok) divided by sentences and with one word per line, except for the multi-unit terms.

4.3. Part-of-speech tagging

The part-of-speech tagging is the process of assigning a part-of-speech like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence. The input of a tagging algorithm is a string of words of a natural language sentence and a specified tag set (a finite list of Part-of-speech tags).

This work uses a customized version of Treetagger (Schmid, 1994). This system receives as inputs the file to be tagged and a compiled version of the lexicon, that is a file in which each line contains a word form, a tab character and a sequence of tag/lemma-pairs separated by blanks, such as

```
aback  RB  aback
abacuses  NNS  abacus
abandon  VB  abandon VBP abandon
abandoned  JJ  abandoned VBD abandon VBN abandon
abandoning  VBG  abandon
```

The CCG-IE lexicon is domain-dependent and includes biological terms. In order to generalize the system, the customized parameter file was replaced by a general English language parameter file which uses the English morphological database. Thus this process is not linked anymore to a specific domain, but uses general English words.

4.4. Semantic tagging

The purpose of this module is to recognize the entities that are relevant to the domain. It substitutes some of the POS tags for more semantically oriented labels, such as org (organism), actv (activation verb), etc.

To carry out the Named-Entity Recognition task, the CCG-IE system uses a dictionary created by the domain experts which contains relevant terms; this is, a list of biological terms including names for genes, proteins, and other physical entities, and also verbs and other nouns. To make the system as general as possible, this domain-oriented dictionary was mapped to the ontology as explained in Section 3.3.

To perform the semantic tagging process, the system gets the names of all the biological objects that take part in a regulatory network; in other words, all the ontology instances (including their synonyms) of the different subclasses of the PhysicalObject class (see Fig. 9). This gives as a result a total of 16799 biological entities. A similar process is performed with the 31689 instances of the Biological subclass; as explained in Section 3.3 this class contains the other part of the domain lexicon; that is, biologically- oriented words that are not names of biological entities. At the end, the instances of the PhysicalObject and Biological classes form a domain

Some of the biological terms are shown in Fig. 9. Instances of the PhysicalObject subclasses.
lexicon of 48488 terms with an associated tag. This lexicon is fed into a script that generates a compiled version of the lexicon, and uses it to tag the matching words within the analyzed text.

4.5. Parsing

The output file from the semantic tagging module is fed into the Abney’s CASS/SCOL parser (Abney, 1996) that uses the CASS cascading grammar parser to generate a tree-like structure by applying a grammar focused on the expression of biological concepts. This markup allows for the recognition of biological entities and the detection of relationships that can be inferred from the grammatical structure of the sentences. The core CASS grammar was adapted from the one developed for the String-IE system for transcriptional regulation (Section 2.2.4).

In the new version of the system, two files have been added to this step: a modified grammar that is focused on the expression of biological concepts and a dictionary of tags that serve as a tag map file to unify the POS tags of the input file with the ones that the CASS grammar uses. These two inputs have been included in the ontology through instances of the Grammar class in order to keep all the domain knowledge independent from the developed OBIE system.

The parsing stage is divided in two different parts: tag set mapping and grammar identification.

4.5.1. Tag set mapping

The CASS parser expects its input file to be already tagged, so it is necessary to adapt CASS to use the output of any tagger that uses a standard tagset. This process is called tag set mapping and uses the UPenn Treebank tagset corpus. The CASS input should consist of tab-separated fields, with one word per line of input (just like the output of the semantic tagging module). The first field contains a word and the second field contains a part of speech. The part of speech that CASS uses depends on the grammar. To avoid retagging with every change in the grammar, the tag mapping is done to map from a standard tag set to the grammar-specific tag set. The mapping is specified by a tag map file which is recognized by CASS as .fx file like the following:

```
aa   ala   *
aa   alanine  *
aa   arg   *
aa   arginine  *
according   according vbg
actdom   activating  *
by   through IN
regul   control NN
regul   regulation  *
```

This tag map converts the UPenn tagset into the tagset of the specific grammar. The file consists of mappings. As with CASS input files, the fields are tab-separated. The mapping “y w x” should be read “output new tag y wherever word w appears with old tag x”. Either the word or the old tag can be wildcarded with *. For instance, looking at the previous example, each time that the word “control” appears with the tag NN, the system should change this tag to “regul”.

This .fx file was built by the system by taking the instances of the ParseTerms class and putting them in the format described above. As shown in Fig. 10, the term “unsequenced” creates the tuple, jj unsequenced /C3. After the 23,301 instances of ParseTerms were placed on the map file, the SCOL parser compiled the .fx file creating the .fxc file which is the one used by the grammar.

4.5.2. Grammar application

Since CCG-IE is a domain specific system, the default CASS grammar is not enough for parsing, that is why a grammar focused
on the expression of biological concepts is used. In order to gen-
eralize the system, this grammar was included as part of the ontol-
ogy as explained in Section 3.4. When the parsing process is
performed, the system extracts all the grammar rules from the
ontology and processes them to build a .reg file. This file is required
by SCOL, together with the .fxc file explained in the previous
section.

The SCOL parsing process is briefly described below because it is
helpful to understand the format of the .reg file in which the ontol-
ogy instances were placed. CASS consists of a pipeline of special-
ized recognizers organized by levels. At the lowest level (level 1)
the input consists of words with parts of speech. Level 2 finds all
sequences at level 1 that match a given pattern (e.g. date expres-
sions) and it reduces those sequences to single elements with the
appropriate category (e.g. Date). The output of level 2 then be-
comes the input to level 3 and so on. This grammar has 4 levels;
thus, the compilation process produces 4 automata and the parser
runs these automata in a cascading manner, one after the other.

4.6. Interaction selection

This last module selects the regulatory interactions. The CCG-IE
system converts the parsed file into an XML format (Fig. 11) and
processes it using customized heuristic modules that:

(a) Identify the regulatory interactions that are to be extracted
and, when possible, the kind of the interaction extracted
(activation or repression).

(b) Create an XML output file with a regulatory network
retrieved from the processed raw text (Fig. 12).

---

**Fig. 11.** Part of the cascaded file converted into a XML file.

**Fig. 12.** Part of the final XML format with the identified interactions.
This process is very attached to the domain knowledge since the heuristics in this module were designed very carefully by human experts to detect and extract a set of specific regulatory expressions between biological entities. This step has not yet been generalized. At the moment, once the grammar has been applied, the domain expert should customize this interaction extraction module to extract the particular domain relationships in the preferred format. Using XML is not mandatory.

5. Case studies: E. coli network regulatory interactions

This section shows the results obtained after testing the system with four different documents. The corpus used to measure the system performance was obtained from the PubMed website and it consists of a set of two full-papers and two abstracts.

5.1. Abstracts

The following subsections detail the results obtained through the analysis of the two abstracts (see Table 3, ids. 1 and 2).

5.1.1. Abstract 1

In the case of abstract 1, the main goal is to detect the regulatory expression embedded in the last sentence (“Furthermore, we show that expression of fimB and fimE is strongly influenced by the H-NS nucleoid protein”).

Fig. 13 shows the parsed file resulting from the analysis of the last three sentences of the abstract. In this case, the bold text emphasizes the parsing result of the identified interactions.

The attributes that define the identified interactions and their participants have been assigned, e.g.:

(a) Attributes for an identified interaction:
- ID: 1
- Form: ev_reg_expr_vp
- Function: activator

(b) Attributes for the participants:
- Regulator H-NS
  - ID: ECK120000450
  - Organism: E. coli
  - Type: nxprot (protein)
- Regulated fimB
  - ID: ECK120000303
  - Organism: E. coli
  - Type: nxpg (gene)

The last step was just to create single entries for all unique interactions in the previous file and give some format to the final XML-like output, shown in Fig. 14. For this case of study the results were completely satisfactory since the two existing interactions were found by the system.

5.1.2. Abstract 2

In this occasion, the system successfully identifies three interactions:

(1) The first interaction corresponds to the sentence: “The expression of the putP-lacZ gene was activated by the glnG gene product and the catabolite gene activator protein (CAP).”
- Interaction ID="1" from="ev_act_expr_vp" ri_function="activator"
- Regulator: glnG GenProtID="ECK120000378" type="nxgene"
- Regulated: putP GenProtID="ECK120000793" type="nxgene"

(2) The second interaction is enclosed in: “The expression of the putA-lacZ gene was activated by CAP and repressed by the glnG gene product.”
- Interaction ID="3" from="anaph+ev_act_expr_xr" ri_function="repressor"
- Regulator: glnG GenProtID="ECK120000378" type="nxgene"
- Regulated: putA GenProtID="ECK120000792" type="nxgene"
Fig. 13. Part of the final parsed file of abstract 1 with the identified interactions.

<table>
<thead>
<tr>
<th>xml version=&quot;1.0&quot;</th>
<th>&lt;interaction ID=&quot;1&quot; from=&quot;ev_reg_expr_vp&quot; ri_function=&quot;activator&quot; source=&quot;XXXX&quot;&gt;</th>
<th>&lt;interaction ID=&quot;2&quot; from=&quot;ev_reg_expr_vp&quot; ri_function=&quot;activator&quot; source=&quot;XXXX&quot;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;regulatory GenProtID=&quot;ECK120000450&quot; org=&quot;ecoli&quot; type=&quot;nprot&quot;&gt;</td>
<td>&lt;regulatory GenProtID=&quot;ECK120000306&quot; org=&quot;ecoli&quot; type=&quot;nprot&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;fimb&gt;</td>
<td>&lt;fimb&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;fimE&gt;</td>
<td>&lt;fimE&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;evidence verb=&quot;influenced&quot;/&gt;</td>
<td>&lt;evidence verb=&quot;influenced&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;sentence pmid=&quot;XXXX&quot;/&gt;</td>
<td>&lt;sentence pmid=&quot;XXXX&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>Furthermore, we show that expression of fimb and fimE is strongly influenced by</td>
<td>Furthermore, we show that expression of fimb and fimE is strongly influenced by</td>
</tr>
<tr>
<td></td>
<td>the H-NS nucleoid protein.</td>
<td>the H-NS nucleoid protein.</td>
</tr>
<tr>
<td></td>
<td>&lt;/sentence&gt;</td>
<td>&lt;/sentence&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;interaction&gt;</td>
<td>&lt;interaction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;interaction ID=&quot;1&quot; from=&quot;ev_reg_expr_vp&quot; ri_function=&quot;activator&quot; source=&quot;XXXX&quot;&gt;</td>
<td>&lt;interaction ID=&quot;2&quot; from=&quot;ev_reg_expr_vp&quot; ri_function=&quot;activator&quot; source=&quot;XXXX&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;regulator GenProtID=&quot;ECK120000450&quot; org=&quot;ecoli&quot; type=&quot;nprot&quot;&gt;</td>
<td>&lt;regulator GenProtID=&quot;ECK120000306&quot; org=&quot;ecoli&quot; type=&quot;nprot&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;fimb&gt;</td>
<td>&lt;fimb&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;fimE&gt;</td>
<td>&lt;fimE&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;evidence verb=&quot;influenced&quot;/&gt;</td>
<td>&lt;evidence verb=&quot;influenced&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;sentence pmid=&quot;XXXX&quot;/&gt;</td>
<td>&lt;sentence pmid=&quot;XXXX&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>Furthermore, we show that expression of fimb and fimE is strongly influenced by</td>
<td>Furthermore, we show that expression of fimb and fimE is strongly influenced by</td>
</tr>
<tr>
<td></td>
<td>the H-NS nucleoid protein.</td>
<td>the H-NS nucleoid protein.</td>
</tr>
<tr>
<td></td>
<td>&lt;/sentence&gt;</td>
<td>&lt;/sentence&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/interaction&gt;</td>
<td>&lt;/interaction&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/RegulatoryNetwork&gt;</td>
<td>&lt;/RegulatoryNetwork&gt;</td>
</tr>
</tbody>
</table>

Fig. 14. Final XML format showing the interactions found in the text.
Finally, the last interaction is also included in this sentence: "The expression of the putA-lacZ gene was activated by CAP and repressed by the glnG gene product."

- Interaction ID="2" from="ev_act_expr_vp" ri_function="activator"
- Regulator CAP GenProtID="ECK1200000160" type="nxpg"
- Regulated putA GenProtID="ECK120000792" type="nxgene"

### 5.2. Full-text analysis

Due to the length of the full text papers, images of the output results are not included in this work; instead, a comparison between the expressions found by the system and the ones found by an expert is presented below.

According to the results summarized in Table 4, it can be inferred that the system performs considerably well finding interactions, as it detected all the regulatory expressions that the expert did. It just needs to be modified to avoid detecting the same interaction more than one time.

### 6. Conclusions and future work

As mentioned at the beginning of this paper, in recent years Information Extraction has become very active in the field of Bioinformatics. This work gave an overview of the current state of IE applied to the Molecular Biology domain, where most of the developed systems are rule-based and others combine statistical methods and rules. The implemented system was meant to be a domain-independent extension of the CCG-IE system.

This work was aimed to generalize the IE system proposed by the CCG and to improve its performance in terms of scalability and reusability by implementing new techniques of ontology-based Information Extraction (OBIE).

The new OBIE system is based on the use of a novel ontology, that was designed from the analysis of the structure of RegulonDB. The ontology was also automatically populated with the biological entities from RegulonDB by developing a methodology to translate the RegulonDB entity-relation structure to an ontology structure as explained in Section 3. Generic mechanisms for Information Extraction where applied to exploit the semantic content of the information in the created domain ontology. Finally, four cases of study (consisting of two abstracts and two full text papers) were used to test the system's performance. The interactions that the system found were compared with the ones that a human expert found in the same corpus, obtaining very good results. As in the case of all unsupervised IE systems, it will always be necessary that a human expert analyzes the obtained results to check their correctness. However, this process is much less time consuming that the current practice of manual analysis of the full content of each paper.

As for future work, an important aspect is to test the system with a bigger corpus consisting in both abstract and full-text documents in order to have a more realistic idea of the performance of the system, and also to measure its accuracy (precision) and coverage (recall).

Another important improvement is to find a way to generalize the last module of the system, the interaction selection, so that the system could be completely domain independent and the user would only need to provide the domain ontology without modifying anything from the system's code. Then, the system could be tested in other domains of knowledge.

### Acknowledgments

This work has been supported by the Spanish Ministry of Science and Innovation (DAMASK project, Data mining algorithms
with semantic knowledge, TIN2009-11005) and the Spanish Government (Plan E, Spanish Economy and Employment Stimulation Plan). A. López was supported by a Fundación Carolina research Grant.

References


