# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities




WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us?
## Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



16

# Biomedical Literature Mining for Biological Databases Annotation

Margherita Berardi [1,5], Donato Malerba[1] , Roberta Piredda[2], Marcella Attimonelli[2], Gaetano Scioscia[3,4] and Pietro Leo[3,4]
[1]*Dipartimento di Informatica – Università degli Studi di Bari*
[2]*Dipartimento di Biochimica e Biologia Molecolare "E.Quagliariello" – Università degli Studi di Bari*
[3]*IBM Italia S.p.A. - Molecular Biodiversity Laboratory*
[4]*IBM Italia S.p.A. - GBS Innovation Centre*
[5]*Exhicon S.r.l., Bari*
*Italy*

## 1. Introduction

In biological research, there are thousands of specialized data repositories, focusing on particular molecules, organisms or diseases, which offer sets of richly annotated records. To ensure data of the highest quality, manual data entry and curation (annotation) processes are generally performed on these databases. Database curators are domain experts who search biomedical research literature for facts of interest, and manually transfer knowledge from published papers to the database. This helps experts to consolidate data about a single organism or a single class of entity, often in conjunction with sequence information. Most importantly, this process makes the information searchable through a variety of automated techniques, given that the curators use standardized terminologies or ontologies. However, as the volume of biomedical literature increases, so does the burden of curation, making annotation databases incomplete and inconsistent with the literature. It has been shown empirically that manual annotation cannot keep up with the rate of biological data generation (Baumgartner et al., 2007). Seemingly, simple tasks of gene annotation by means of a controlled vocabulary becomes very laborious since an expert is required to inspect carefully the whole literature associated to each gene, to identify the appropriate terms. On the other hand, the contribution of the manual annotation community is essential to the understanding of the ever more complicated biological landscape and it is widely accepted that it produces the most accurate annotations currently available. To reduce the cost of obtaining annotations, several initiatives for collaborative curation, such as community annotation projects (e.g., http://www.pseudomonas.com/) and wiki-based prototypes (e.g., http://www.wikiprofessional.org/) have been recently promoted. Nevertheless, there is not enough evidence to clearly assess if collaborative curation solves the problem (Lu et al., 2007). As of now, PubMed remains the richest and most updated source of information about biological data despite its unstructured nature. This motivates the upsurge of interest in text mining techniques which enable various degrees of automation in the analysis of

Open Access Database www.intechweb.org

Source: Data Mining in Medical and Biological Research, Book edited by: Eugenia G. Giannopoulou,
ISBN 978-953-7619-30-5, pp. 320, December 2008, I-Tech, Vienna, Austria

scientific literature, such as identification of named entities, classification of documents, extraction of relevant facts (i.e., relationships between two or more named entities expressing a fact), and generation of hypotheses (Cohen & Hersh, 2005; Jensen et al., 2006; Krallinger et al., 2005). The fundamental challenge in the application of data mining to text data is the translation of text into such a structured form that should, intrinsically, encapsulate the data semantics.

Despite the fact that many text mining systems have been deployed in the biomedical context and reasonable levels of performances on gold standard data have been achieved (Hirschman et al. 2005; Cussens and Nedellec, 2005; Shatkay and Feldman, 2003), the actual contribution to database curation efforts is still unclear (Yeh et al., 2003; Rebholz-Schuhmann et al., 2005). Most of these systems have been developed to solve very specific problems on task-tailored data and very few of them have been concretely used to assist curators in the population of biological databases. An example in this direction is the PreBIND system (Alfarano et al, 2005) which serves to curate the BIND (http://bind.ca/) protein-protein interactions database. This uses a combination of statistical methods for relevant document retrieval and rule-based methods for bio-molecule name recognition with the aim to find statements about protein interactions. It is reported that the system is able to reduce the time necessary to perform a representative task by 70%, savings 176 person days thanks to its ability to suggest candidate additions. Similar example is the LSAT system (Shah and Bork, 2006) developed for the extraction of alternative transcripts to populate the ASD database (http://www.ebi.ac.uk/asd). The MuteXt system (Horn et al., 2004) extracts from literature point mutations useful for the maintenance of the GPCRDB (www.gpcr.org/7tm/mutation/) and the NucleaRDB (www.receptors.org/NR/mutation/) protein databases. LSAT performs automatic classification of sentences about transcripts and automatic role labelling of text tokens. MuteXt exploits manually encoded regular expressions to capture textual patterns. In both cases, a considerable additional effort is necessary since extracted knowledge requires to be manually combined with sequence and structural information. In fact, the nomenclature adopted for entries in a database often uses wording that is very different from what is explicitly stated in text passages; it is also possible that the information to be extracted has to be deduced from more than one portion of text. Links to entries on different databases should be disambiguated and added to make the annotation result useful for data analysis. These aspects further complicate the feasibility of involving completely automatic tools for database annotation. It appears clear that text mining technology can contribute to this field by operating together with curators to minimize their involvement and speed up the pace of research, but it will not completely substitute their role.

In this work, we tackle the problem of supporting biological database annotation through a data mining approach to Information Extraction (IE). IE is the discipline that aims to extract relevant information from natural language documents. The goal of an IE process is to map unstructured text into structured form, such as databases or knowledge bases, by filling pre-specified information templates describing objects of interest (i.e., entities such as a protein or, more specifically, a kinase) and facts about them (e.g., phosporylation or interaction relationships). This is achieved by supplying quite sophisticated language processing methodologies (e.g., taggers, chunkers, light semantic interpreters, information extraction rules) and domain-specific resource developments (e.g., dictionaries and ontologies). While significant progresses have been made in developing tools for IE from biomedical data, the

difficulties encountered in adapting systems to new applications and domains remain the main barriers to their wider use. Thanks to their ability to analyse large volumes of unstructured data, data mining methods are promising candidates to alleviate the burden in developing and customizing IE systems to extract the required domain-specific knowledge.
More precisely, we address the problem of mining extraction patterns for information template filling, i.e., to discover conditions to fill slots of templates of interest. Domain experts are asked to define annotation schema in terms of entities and templates (i.e., a set of properties characterizing each entity) and to provide examples of documents labelled with filled templates. Discovered patterns allow the IE system to automatically identify template instances occurring in new documents. We describe a strategy for extraction pattern mining which is based on an Inductive Logic Programming (ILP) approach to recursive theory learning from examples. It is implemented in the ATRE[1] system which works on logical representations of the textual content. Implemented methods are general and domain-dependency is limited to specific thesauri of the biomedical domain. We present a real-world case study concerning the annotation in HmtDB[2] of mitochondrial (mt) DNA. HmtDB stores human mt genomes from healthy or pathological phenotypes and their variability and clinical data associated to diseases are annotated (Attimonelli et al., 2005).

## 2. Background

Text mining tasks for biomedical literature mining can be grouped into some few main classes (Jensen et al., 2006; Shatkay & Feldman, 2003; Cohen & Hersh, 2005). First, *named entity recognition* aims to identify, within a collection of text, all of the instances of a name for a specific type of thing. The detection of biologically significant entities such as gene and protein names is a very important task for biological database curation since these constitute the main entry points for biological databases. Second, *text classification* attempts to determine automatically whether a document or part of a document discusses a given topic or contains a certain type of information. Accurate text classification systems can be especially valuable to database curators, who may have to review many documents to find a few that contain the kind of information they are collecting in their database. Third, *terminology extraction* aims to collect synonyms and abbreviations of biomedical entities to aid literature search engines and mining systems to be more precise. Fourth, *relationship extraction* systems detect occurrences of a pre-specified type of relationship between a pair of entities of given types. Finally, *hypothesis generation* (Srinivasan, 2004) attempts to uncover relationships that are not present in the text but may be inferred by the presence of other more explicit relationships (e.g., if "BRCA1" and "breast cancer" occur in the same sentence, a relationship between breast cancer and the BRCA1 gene might be assumed).
In the IE literature for biomedicine, little attention has been devoted to classic IE tasks of template filling (Gaizauskas and Wilks, 1998), despite the fact that these naturally fit in database annotation problems. For instance, considering the annotation schema adopted to develop and maintain the IARC TP53 database (http://www-p53.iarc.fr/Help.html#annotations) which compiles all TP53 mutations that have been reported in the

[1] http://www.di.uniba.it/~malerba/software/atre
[2] http://www.hmtdb.uniba.it

published literature since 1989, we can observe that main annotations (i.e., mutation, tumour, demographic information, reference and detection method) are structured in form of templates. Each template correlates some entities (e.g., the detection method is added to the database by collecting information on tissue processing, start material, pre-screening method, sequenced material, etc.). Instances of each template can be extracted from a paper pertaining TP53 mutations by analysing relationships implicitly expressed to link target entities. While in a named entity recognition task, the goal is to identify peculiar objects of interest, such as all the disease names occurring in a text, in a template filling task, conceptual relationships between named entities, such as the DNA position and the mutant base pairs characterizing a mutation, should be taken into account.

Several strategies ranging from hand-coded patterns to various machine learning based approaches have been employed to solve this class of problems (Nédellec, 2004). In this work, we follow a different strategy based on the remark that template filling tasks, which are generally based on the results of a named entity recognition task, can be simplified when tagging of named entities is, in its turn, performed by considering conceptual dependencies implicitly defined at either the syntactic or structural level (e.g., the type of mutation is normally reported before the DNA position). Therefore, we adopt a method to learn tagging models in the form of recursive logical theories which can naturally represent conceptual dependencies between named entities. We report results of a first tentative to annotate HmtDB data related to human mtDNA mutations in diseased phenotypes. Thus, the issue is to extract from relevant papers information regarding the mutation and the features associated to the phenotype.

## 3. Issues

Recursive theory learning falls within the class of supervised concept learning methods, which are supplied with information about objects whose class (or concept) membership is known (i.e., training examples) and produces from this a characterization of each class in some formal language. If U is a universal set of objects (or observations), a concept C can be formalized as a subset of objects in U: $C \subseteq U$. To learn a concept C means to learn to recognize objects in C.

***Inductive concept learning***. Given a set E of positive and negative examples of a concept C, find a hypothesis H, expressed in a given concept description language L, such that every positive example is covered by H and no negative example is covered by H.

In Inductive Logic Programming (ILP) (Muggleton, 1992; Nienhuys-Cheng and de Wolf, 1997) the formal languages for describing objects and concepts are typically based on Horn clausal logic. More precisely, concepts to be learned are represented by means of predicate symbols, and the result of the learning process is a logical theory. In the IE framework considered in this work, concepts to be learned correspond to entities involved in a template of interest and the logical theory includes clauses expressing the conditions to fill template slots.

The typical formalism adopted in ILP allows the representation of relational (or structural) patterns. In particular, classification rules can express conditions on both properties of single objects and relations between them. In addition classification rules can also express dependencies or relations between concepts. This is a main issue in information extraction from biomedical text since it is the typical application where examples, in addition to their inherent relational structure, present relations to other examples. Some authors have already

used ILP to construct theories for information extraction (Aitken, 2002; Goadrich et al., 2004). In particular, the work by Goadrich et al. (2004) tackles the problem of learning biomedical target relationships (i.e., protein-location) between items of text, namely multi-slot extraction (i.e., two-slot extraction). Our goal is to learn single-slot extraction rules that should take into account implicit relations expressed in the text between entities of the same template. For this reason, we resort to recursive theory induction as learning framework, since recursive theories can express well-defined mutual dependencies between predicates. A different IE problem is handled with ILP in (Ramakrishnan et al., 2007), that is automatic feature construction. The authors employ ILP to define new features given a logical representation of texts and some background knowledge. This is an important problem since one of the issues in IE concerns the definition of the appropriate representation of text. Afterwards, additional issues are raised by the complexity of text processing operations necessary to produce logical representations of textual content. Several sources of difficulties are peculiar of the biomedical language such as ambiguities occurring when the same term denotes more than one semantic class (e.g., p53 is used to specify both a gene and a protein) or when many terms lead to the same semantic class (abbreviations, acronym variations); continuous creation of new biological terms or evolutions of the same biological object (e.g., genes are renamed once their function is known); use of non standard grammatical structures as well as domain-specific jargon; gene symbol polysemy (i.e., a symbol can refer to more than one gene, both within a single species an disparate organisms). This makes the preparation of training data really difficult. A number of controlled vocabularies, lexicons and ontologies for biomedicine which can be exploited both in the data preparation and reasoning steps are available. This further motivates an ILP approach which can naturally handle external background knowledge.

In the rest of the chapter we briefly introduce the HmtDB resource and the information extraction problem involved in curation activities. Our approach to training data preparation and rule learning is proposed. A framework which integrates the proposed solution to support experts in the training of the mining module and to revise annotation results is described.

## 4. The HmtDB annotation case study

### 4.1 The biomedical problem

Mitochondrial DNA (mtDNA) has been widely studied both in population genetics and mitochondrial disease studies. In particular, the high mutation rate, absence of recombination, and maternal transmission all make this DNA different from its nuclear counterpart and suitable for evolutionary studies aimed at tracing the migrations which led to the colonization of the various geographic areas of the world. The mtDNA genome of two unrelated individuals may differ in the presence of about 50 mitochondrial Single Nucleotide Polymorphisms (mtSNPs) (Wallace D. C. et al. 1999; Smeitink J. et al., 2001). Study of these polymorphisms in various human populations has allowed us to group differing human mtDNAs in haplogroups, each containing a subset of mtDNA sharing characteristic mutations acquired from the same ancestral mtDNA molecule. Hence, various population lineages may be described by means of a phylogenetic network, in which the top nodes define haplogroups and the tips define haplotypes represented by the sequence of the entire mitochondrial genome in the best situation (Torroni A. et al., 2001). Nevertheless,

mitochondrial DNA also plays an important role in the oxidative metabolism of the cell. Hence, mutations occurring in mitochondrial DNA can alter the oxidative phosphorylation, which seriously damages cells and tissues, causing mitochondrial diseases. Mitochondrial disorders - associated with dysfunctions of the Oxidative Phosphorylation (OXPHOS) system - are caused by genetic defects both in the mitochondrial and nuclear genome, leading to energy metabolism errors, and have an estimated frequency of 1 out of 10000 live births. Due to the important role played by the OXPHOS system in ATP production, the causes and effects of mitochondrial disorders are extremely heterogeneous and complex. This explains the pressing need for further research on this topic, despite the many studies on mitochondrial disorders published in the last 20 years. In this scenario HmtDB (Attimonelli M. et al., 2005) plays an important role, gathering all complete human mitochondrial genomes worldwide distributed and enriching sequence information with statistically validated variability data estimated through the application of specific algorithms implemented in an automatically running Variability Generation Work Flow (VGWF). Knowledge through HmtDB of the variability of specific position of the genome is highly informative, as shown in a recent study by Accetturo et al. (2006), which demonstrates that continent specific high variability values can act as haplogroup markers.

### 4.2 Database description

HmtDB consists of a database of Human Mitochondrial Genomes annotated with population and variability data, the latter estimated through the application of a new approach based on site-specific nucleotidic and aminoacidic variability calculation (Pesole & Saccone, 2001; Horner & Pesole, 2003). Currently, HmtDB stores data from entire human mt genomes only, while a great quantity of published data related to single human mtDNA mutations and associated to clinical studies available through PubMed are not annotated in HmtDB.

In particular, HmtDB

- collects and integrates the publicly available human mitochondrial genomes data;
- produces and provides the scientific community with site-specific nucleotide and aminoacid variability data estimated on all the collected human mitochondrial genome sequences;
- allows all researchers to analyse their own human mitochondrial sequences (both complete and partial mitochondrial genomes) in order to automatically detect the nucleotide variants compared to the revised Cambridge Reference Sequence (rCRS) (Andrews et al., 1999) and to predict their haplogroup paternity.

At present, HmtDB contains 4061 human mitochondrial genomes. They are stored and analysed as a whole dataset and grouped into continent-specific subsets (AF: Africa (347 mtGenomes), AM: America (216), AS: Asia (1493), EU: Europe (1233), OC: Oceania (133)); 729 genomes are unclassified as regards the geographic origin of individual donors of DNA. Human mtDNA is composed by 16569 nucleotides, of which 4542 (data from HmtDB) present variability values differing from 0.

HmtDB can be queried according to different criteria combined among them through the AND Boolean Operator. The most important selection criteria are listed in table 1.

Multi-alignment and site-variability analysis tools included in HmtDB are clustered in two workflows: the Variability Generation Work Flow (VGWF) and the Classification Work Flow (CWF), which are applied both to human mitochondrial genomes stored in the database and to newly sequenced genomes submitted by users, respectively.

| Selection criteria in HmtDB | Meaning of criteria |
|---|---|
| Subjects' geographical origin | Continent and Country origin of the subject |
| Haplogroup Code | Code assigned by population genetists to human mtDNA genomes clustered according to common mtSNPs |
| SNP Position | Position of the genome where variation is sought |
| Variation type | Transition, trasversion, deletion, insertion |
| Subject Age (year) | Age of the subject when DNA was extracted |
| Subject Sex | Sex of the subject donor of the DNA |
| DNA source | Blood, tumor tissue, buccal swab, blood, etc |
| Individual type | Returns genomes correlated to the selected phenotype: Normal, Patient, Control or Disease Phenotype |
| References | Journal, Authors, Haplotype paper code, PubMedID |

Table 1. Database search criteria

### 4.3 Database annotation

HmtDB currently stores data derived from the knowledge of complete mt genomes. 635 out of the 4061 genomes stored in HmtDB are related to disease phenotypes associated to 13 different diseases. This last datum highlights the real need of the experience presented here. The number of mt diseases is far higher than 13, and literature reports data from single mt mutations screened in families and populations to assess associations of mutation with mt diseases. This type of information is available through MITOMAP[3], but here the mtSNP is associated with the various phenotypes and literature data in a qualitative way and thus the data structure does not allow any quantitative estimate of the occurrence of the mutation in different phenotypes and different populations. Our goal is to include in HmtDB data extracted from the great quantity of papers available on the topic "mtDNA mutation and disease" and to integrate the data, structured and analysed with statistical tools, with the variability data derived from the human mt complete genomes already in HmtDB, thus allowing a comprehensive study of mtDNA variability related to population genetics and mt diseases. Until now, this has been partially carried out through manual inspection of mitochondrial literature. We are currently testing the text mining approach proposed in this work to perform automatic extraction of information from PubMed. Desired information concerns mutations, method of detecting mutations, and demographic details. More precisely, the nature of a mutation (e.g, insertion, deletion, transversion, etc.), mutant base (i.e. nucleotide), involved gene (i.e., locus), type of pathology, age, sex and nationality of patients/individuals, method and source of analysis (e.g., tissue, blood, etc.) are all categories of terms to be automatically identified in texts.

### 4.4 Literature preparation

Generating a reliable training set is a slow and labour-intensive task since there are no publicly available datasets about mtDNA annotation from the literature. For this aim,

[3] http://www.mitomap.org/

HmtDB curators have performed several steps: (1) retrieval of relevant literature, (2) definition of annotation schema, (3) collection of domain-dependent language resources, (4) manual annotation of text. The first and the last steps were the most demanding. To retrieve literature pertaining to the HmtDB scope, PubMed was queried for "mitochondrial disease", looking for papers published after 2000 concerning human mtDNA and where mutations involving mitochondrial genes were studied. Papers that do not report strictly clinical data were discarded. All papers in which mitochondrial mutations and diseases were associated on the basis of alternative information such as biochemical experiments, drug treatment or therapy, species different from the human type, etc. were also excluded from the dataset as well. Then, a suitable annotation schema was defined. Two main schema appear to meet the database annotation requirements: one to describe features reported on the *mutation* and the other describing *subjects* from whom the DNA comes. As shown in table 2, the mutation template includes ten categories of information, while the subjects template involves seven.

| Template | Entity | Definition |
|---|---|---|
| mutation | heteroplasmy | Quantity of mutant mtDNA (%) |
| | locus | Gene where mutation is found |
| | novelty | Flag stating that mutation is published for the first time |
| | pathology | Disease or syndrome (phenotype) |
| | penetrance | Different pathological phenotype expression |
| | position | Nucleotide position in mtDNA where mutation is located |
| | risk | Probability of expressing a pathological phenotype |
| | substitution | Nucleotide changed, compared with reference sequence (rCRS) |
| | type | Type of mutation |
| | type_position | Type and position encoded by a single alphanumeric string |
| subjects | age | Age of the subject |
| | category | Single patient or pedigree |
| | gender | Gender of subject |
| | method | Biomolecular method to detect mutation |
| | nationality | Geographic origin |
| | number | Number of subjects affected |
| | source | Biological source of mtDNA |

Table 2. Database annotation schema

Domain-dependent dictionaries to guide the annotation process are carefully selected by curators. Some dictionaries are obtained by directly extracting controlled vocabularies stored in the database, like methods, diseases, ethnic groups, locus and sources.

In the final step, selected papers were manually analysed to identify pieces of text satisfying the annotation schema. This was the most laborious phase, as curators are asked to perform manual tagging in the most homogeneous way by inevitably facing different difficulties due to the non-standard way of publishing information. A first problem is raised by gene referencing, since each gene typically has several names and abbreviations (e.g., the ATP6 mt gene can be also mentioned as "ATP synthase F0 subunit 6" or "MTATPase6" or "adenosine triphosphatase 6" as well) and sometimes authors and publishers do not agree

on standards. A mt gene abbreviation dictionary has been prepared by curators to give direction to a locus annotation strategy. However, the most complex activity concerns the identification of the pathology associated with the mutation. This results to be very hard because many different clinical presentations of mitochondrial diseases are possible. Hence, the diagnosis is often not really established by the authors themselves, while one or more terms are used to describe a large collection of disorders. For instance, in the following abstract:

*"The authors describe a novel pathogenic G5540A transition in the mitochondrial transfer RNA (tRNA)Trp gene of a sporadic encephalomyopathy characterized by spinocerebellar ataxia. Clinical features also included neurosensorial deafness, peripheral neuropathy, and dementia"*

the pathological condition is defined by a standard "encephalomyopathy" disease with additional symptoms such as "spinocerebellar ataxia", "neurosensorial deafness", etc. In addition, typical mitochondrial disease names are obtained by grouping different symptoms (e.g., the "MELAS" syndrome is considered in the case of "mitochondrial myopathy, encephalopathy, lactic acidosis, and strokelike episodes") but a standard nomenclature of mitochondrial diseases is not available in the scientific community; in fact, there are also cases, as in the abstract reported below, where atypical combinations of symptoms are connected to the mutation.

*"Mitochondrial cytochrome b mutations have been reported to have a homogenous phenotype of pure exercise intolerance. We describe a novel mutation in the cytochrome b gene of mitochondrial DNA (A15579G) associated with a selective decrease of muscle complex III activity in a patient who, besides severe exercise intolerance, also has multisystem manifestations (deafness, mental retardation, retinitis pigmentosa, cataract, growth retardation, epilepsy)".*

To manage such cases, the MITOMAP annotation of diseases associated with mtDNA mutations from the perspective of phenotype was adopted as a reference. There are also some papers describing not only a patient but their family pedigrees, which lead to very heterogeneous clinical presentations also creating coreference resolution problems. E.g., *"The proband showed isolated, spastic paraparesis. A brother, who had suffered from a multisystem progressive disorder, ultimately died of cardiomyopathy. Another brother is healthy. The proband's mother showed truncal ataxia, dysarthria, severe hearing loss, mental regression, ptosis, ophthalmoparesis, distal cyclones, and diabetes mellitus. ... Sequence analysis of mtDNA showed a heteroplasmic mutation of the tRNA(Ile) gene (G4284A). ..."*

The curators decided to consider the abstract and the title of each article, since other mutations and populations that are not being studied are often cited in the introduction and discussion sections of papers. Indeed, selecting relevant portions of text is a prerequisite step for IE, since the sparseness of data and lack of robustness of IE methods makes them inapplicable to large corpora or irrelevant texts.

## 5. The approach

### 5.1 Problem definition

The problem we are addressing is the typical template filling task reported in the IE literature (Gaizauskas & Wilks, 1998). This means that, rather than learning one extraction pattern for each slot of interest, a single model for all slots of interest is learned. Dependencies among facts are also investigated in the context of template filling, since the pattern should link isolated facts in some way.

Let us consider the following example of a text fragment of the collection described in the previous section:
*"Cytoplasts from two unrelated patients with MELAS (mitochondrial myopathy, encephalopathy, lactic acidosis, and strokelike episodes) harboring an A-*G transition at nucleotide position 3243 in the tRNALeU(UUR) gene of the mitochondrial genome were fused with human cells lacking endogenous mitochondrial DNA (mtDNA)"*
Here "MELAS" is an instance of the *pathology* associated to the mutation in question, "A-*G" is an instance of the *substitution* that causes the mutation, "transition" is the *type* of the mutation, "3243" stands for the *position* in the DNA where the mutation occurs, "tRNALeU(UUR)" is the *gene* associated with the mutation, "two" is the number of *subjects* under study. An extraction pattern relating *type* and *substitution* items is exemplified by the following two Horn clauses:

```
substitution(X) ← follows(Y,X), type(Y)
type(X) ← distance(X,Y,3), position(Y),
            word_between(X,Y,''nucleotide position'')
```

The first clause states that a token X is recognized as the *substitution* (i.e., which nucleotide is substituted by which other, A in G in the specimen text) if it is followed by a token Y which has been recognized as mutation *type* (transition). The second clause states that X fills the mutation *type* (transition) slot if it is three words far from a token Y that has been associated with the mutation *position* (3243) slot and there is the intermediate word "nucleotide position".
It should be noted that, in the above example, some dependencies between slots of the same template (mutation) are shown. As previously mentioned, learning information extraction rules which express these dependencies may lead to more accurate models, which reflect some co-occurrence of named entities in the text. In addition, when automated annotation is performed, context-sensitive recognition of named entities is possible, thanks to learned models which reflect dependencies among annotation classes. A solution to the problem of searching for concept dependencies (i.e., mutual recursion) in the space of candidate patterns and to reason in the presence of relational knowledge is provided by the learning algorithm reported in Section 5.4.

### 5.2 Data preprocessing

Texts are preprocessed by means of natural language facilities provided in the GATE (General Architecture for Text Engineering) system (Cunningham et al., 2002). We exploit the ANNIE (A Nearly-New IE system) component which contains finite-state algorithms and the JAPE (a Java Annotation Patterns Engine) language which is also a finite-state transduction engine to recognize regular expressions. We use ANNIE to perform tokenization, sentence splitting, part-of-speech tagging, general purpose named-entity recognition (e.g., persons, locations, organizations) and mapping into dictionaries. We use both predefined dictionaries available with ANNIE (e.g., organization names, job titles, geographical locations, dates, etc.) and domain-specific dictionaries prepared by curators. General domain dictionaries are used to clarify some terms (e.g., places and geographical locations are useful in recognizing terms about the ethnic origin of the diseased sample). Domain-specific dictionaries are flat dictionaries of canonical forms and variants of names of mitochondrial genetics. Some general-purpose biological dictionaries were also considered

e.g., those on enzymes, units of measurement, and nucleic acids. They are exploited to reduce data heterogeneity and to perform syntactic and semantic normalization, such as rough resolution of acronyms which, as already stated, are one of the sources of redundancy and ambiguity. JAPE grammars have been defined to identify appositions occurring in texts as well as some numeric and alphanumeric strings which are frequent in this domain. Lastly, stopwords (e.g., articles, adverbs, and prepositions) are removed and stemming is performed by means of the Porter's algorithm for English texts (Porter, 1997).

### 5.3 Data representation

In this work, the units of analysis are sentences, which are composed of tokens. Each sentence or token is given a unique identifier (in the context of an abstract or a title of selected papers) based on its ordering within the given text. The relational (or structural) representation of a sentence is described by a set of predicates expressing properties of occurring tokens and relations between them.

Properties, which are represented by unary function symbols (or descriptors), express statistical (e.g., token frequency), lexical (e.g., alphanumeric, capitalized token), structural (e.g., structure of complex tokens such as alphanumeric string, abbreviations, acronyms, hyphenated tokens), syntactical (e.g., singular/plural proper/not proper nouns, base/conjugated verbs) and domain-specific knowledge (e.g., an entity belonging to a dictionary). More precisely, the descriptor *class* specifies the category of the described text (i.e., abstract, title, results, etc.) and expresses information on the localization of annotations in documents. The descriptor *word_to_string* maps an identifier to the corresponding stemmed token, while *word_frequency* expresses the relative frequency of a token in the given text, and *type_of* refers to morphological features and takes values in the set {allcaps, mixedcaps, upperinitial, numeric, percentage, alphanumeric, real number}. Parts-of-speech are encoded by the descriptor *type_pos*, and semantics is added by the descriptor *word_category*.

Binary descriptors express structural properties such as the composition of sentences in passages of text and tokens in chunks or directly in sentences. Indeed, the following descriptors have been defined: *part_of*, which lists tokens composing a sentence, and *follows*, which relates a token to its direct successor. Complex tokens (e.g., A-*G) are described by some descriptors (e.g., *middle_is_char*, *first_is_numeric*) defining the morphological nature of an alphanumeric string. Another form of relational knowledge concerns domain dictionaries and expresses the distance between two categorized tokens in the context of a sentence (*distance_word_category*).

For the training data, only sentences containing at least a positive example of concepts to be learned are considered. Henceforth, they are called target sentences. No relation between target sentences is currently considered: that is, the extraction of slot fillers remains local to sentences.

An example of relational description generated for the target sentence reported in Section 5.1 is the following:

```
annotation(3)=no tag,
...
annotation(7)=pathology,
annotation(8)=no_tag,
```

```
...
annotation(13)=substitution,
annotation(14)=type,
annotation(15)=no_tag,
...,
annotation(17)=position,
annotation(18)=locus,
...,
annotation(30)=no_tag
                    ←
class(2)=abstract, part_of(2,3)=true, ..., part_of(2,30)=true,
word_to_string(3)=cytoplast, ..., word_to_string(14)=transition,
..., word_to_string(30)=cell,
type_of(3)=upperinitial, ..., type_of(29)=alphanumeric,
type_pos(3)=nnp, ..., type_pos(30)=nns,
word_frequency(3)=1, ..., word_frequency(30)=2,
word_category(7)=disease, ..., word_category(28)=nucleic_acid,
distance_word_category(7,9)=2, ..., distance_word_category(27,28)=1,
follows(3,4)=true, follows(4,5)=true, ..., follows(29,30)=true
```

It is in form of a multiple-head clause (Levi & Sirovich, 1976), where the body (left) part lists literals describing properties of the sentence and the head (right) part states annotations occurring in the sentence. Constant 2 denotes the described sentence, which belongs to an abstract of the collection. Constants 3, 4, ..., 30 denote identifiers of tokens in the described sentence.

We observe that the particular form of literal used in this work, namely $f(t_1, \ldots, t_n)=Value$, where $f$ is an n-ary descriptor, $t_i$'s are constant terms, and $Value$ is one of the possible values of $f$'s domain, can be easily reported to the typical notation adopted in predicate calculus $p_{f=Value}(t_1, \ldots, t_n)$, where $p_{f=Value}$ is the n-ary predicate associated to the pair $\langle f, Value \rangle$.

Background knowledge is also defined to support qualitative reasoning in the learning phase. This includes a number of Horn clauses such as the following, which express the synonymy between (stemmed) biological terms:

```
word_to_string(X)=transit ← word_to_string(X)=transversion
word_to_string(X)=substitut ← word_to_string(X)=replac
```

A transitive definition of the relation of "indirect successor" was also defined to unburden the representation language, which includes only the direct successor relation:

```
tfollows(X,Y)=true ← follows(X,Y)=true
tfollows(X,Y)=true ← follows(X,Z)=true, tfollows(Z,Y)=true
```

Lastly, a typified form of both direct and transitive successor relations is introduced to compact knowledge encapsulated in rules further. Some examples are reported in the following:

```
follows_string_jj(Y)=Z ← word_to_string(X)=Z, follows(X,Y)=true,
                          type_pos(Y)=jj
follows_nn_string(X)=Z ← type_pos(X)=nn, follows(X,Y)=true,
                          word_to_string(Y)=Z
tfollows_vb_nn(X,Y)=true ← type_pos(X)=vb, tfollows(X,Y)=true,
```

```
                                    type_pos(Y)=nn
tfollows_jj_nn(X,Y)=true ← type_pos(X)=jj, tfollows(X,Y)=true,
                                    type_pos(Y)=nn
```

The first two clauses express the direct successor relations between a generic string and an adjective or a noun, while the last two clauses specify the transitive successor relations for verb-noun and adjective-noun pairs, respectively.

### 5.4 Rule learning

Logical theories used for the annotation of text are automatically induced from training data by means of the ILP system ATRE (Malerba, 2003). In this application, each concept plays the role of an annotation class (i.e., template slot) and each textual object can be associated with at most one concept, i.e., concepts are considered mutually exclusive. The learning problem solved by ATRE can be formulated as follows:

*Given*

- A set of *target* predicates $p_1$, $p_2$, …, $p_r$ to be learned
- A set of positive (negative) examples $E_i^+$ ( $E_i^-$ ) for each predicate $p_i$, $1 \le i \le r$
- A background theory *BK*
- A language of hypotheses $L_H$ that defines the space of hypotheses $S_H$

*Find*

a (possibly recursive) logical theory $T \in S_H$ defining the predicates $p_1$, $p_2$, …, $p_r$ (that is, $\delta(T)=\{p_1, p_2, \ldots, p_r\}$) such that the following two conditions hold:

a. for each *i*, $1 \le i \le r$, $BK \cup T \models E_i^+$ (*completeness* property) and
b. $BK \cup T \not\models E_i^-$ (*consistency* property).

The logical theory T is a set of first-order definite clauses (Lloyd, 1987), like those reported above. The set of concepts to be learned is defined by means of a set of literals of the type annotation(X)=*annotation class*. No clause is generated for the concept annotation(X)=*no tag*. Each unit of analysis, which corresponds to a sentence, is represented by means of the set of positive/negative examples related to the sentence as well as the set of ground literals in the BK which describe properties and relations among tokens in the sentence. The set of literals associated to a unit of analysis is called *object* and is formally represented as a ground (i.e., without variables) multiple-head clause. Therefore, ATRE's representation of training data is individual-centered (Blockeel & Sebag, 2003) and this has both theoretical (PAC-learnability) and computational advantages (smaller hypothesis space and more efficient search).

The background knowledge *BK* may also include a set of Horn clauses which define new predicates, not used for the description of training objects but deemed useful for the formulation of the logical theory used in the annotation process. Examples are the *tfollows* predicates defined in the previous section. An example of Horn clause which defines the predicate *char_number_char* is reported in the following:

```
char_number_char(X) ← first_is_char(X), middle_is_numeric(X),
    last_is_char(X)
```

The satisfaction of the completeness and consistency properties guarantees the correctness of the induced theory with respect to the sets of positive and negative examples, but not necessarily with respect to new instances of the target predicates. The selection of the clause

in T is made on the basis of an inductive bias. For example, clauses which cover a high number of positive examples and a low number of negative examples may be preferred to others.

At high-level, the learning strategy implemented in ATRE is *sequential covering* (or *separate-and-conquer*) algorithms (Mitchell, 1997), that is, one clause is learned (conquer stage), covered examples are removed (separate stage) and the process is iterated on the remaining examples. More precisely, a logical theory $T$ is built step by step, starting from an empty theory $T_0$, and adding a new clause at each step. In this way we get a sequence of theories

$$T_0 = \varnothing, T_1, \ldots, T_i, T_{i+1}, \ldots, T_n = T,$$

such that $T_{i+1} = T_i \cup \{C\}$ for some clause $C$.

The conquer stage aims at finding the best clause $C$ to add. The search for this clause is made among those that cover specific positive examples, called *seeds*, which have not been covered by $T_i$ yet.

The most important novelty of the learning strategy implemented in ATRE is embedded in the design of the conquer stage. Indeed, the separate-and-conquer strategy is traditionally adopted by single predicate learning systems which generate predicate definitions, that is, sets of clauses with the same predicate in the head. In ATRE, clauses generated at each step may have different predicates in their heads. In addition, *the body of the clause generated at the i-th step may include all target predicates $p_1, p_2, \ldots, p_r$ for which at least a clause has been added to the theory $T_i$.* In this way, dependencies between target predicates can be expressed by learned theories.

The order in which clauses of distinct target predicates have to be generated is not known in advance. This means that the actual dependencies between target concepts which a learned theory can express have to be discovered by the system and is not specified by the user. For this reason, it is necessary to generate clauses with different predicates in the head and then to pick one of them at the end of each step of the separate-and-conquer strategy. Since the generation of a clause depends on the chosen seed, several seeds (at least one, if any, per target predicate) have to be chosen among those still uncovered. Therefore, the search space is actually a forest of as many search-trees (called *specialization hierarchies*) as the number of chosen seeds. In each search tree a directed arc from a node (clause) C to a node $C_0$ exists if $C_0$ is obtained from C by a adding a literal (C is specialized into $C_0$).

The forest can be processed in parallel by as many concurrent tasks as the number of search-trees (hence the name of *separate-and-parallel-conquer* for this search strategy). Each task traverses the specialization hierarchy top-down (or general-to-specific), but synchronizes traversal with the other tasks at each level. Initially, some clauses at depth one in the forest are examined concurrently. Each task is actually free to adopt its own search strategy, and to decide which clauses are worth to be tested. If none of the tested clauses is consistent, clauses at depth two are considered. Search proceeds towards deeper and deeper levels of the specialization hierarchies until at least a user-defined number of consistent clauses is found. Task synchronization is performed after that all "relevant" clauses at the same depth have been examined. A supervisor task decides whether the search should be continued or not, according to the results returned by the concurrent tasks. When the search is stopped, the supervisor selects the "best" consistent clause according to the inductive bias specified by the user (e.g., the clause which covers a high number of positive examples and a low number of negative examples). This search strategy provides us with a solution to the

problem of *interleaving* the induction of distinct target predicate definitions. It also has the advantage that simpler consistent clauses are found first, independently of the predicates to be learned. Finally, the synchronization allows tasks to save much computational effort when the distribution of consistent clauses in the levels of the different search-trees is uneven.

A more detailed description of the search strategy implemented in ATRE and its optimization through caching techniques is reported in (Malerba, 2003; Berardi et al., 2004).

### 5.5 The architecture of BEE

The BEE[4] (Biomedical Entity Extractor) system was developed to implement the approach described in the previous sections. BEE supports users in:

- defining annotation schema;
- manually annotating texts to provide mining examples for user classes;
- customizing linguistic analysis through dictionary (gazetteers) management;
- automatically generating data for mining;
- using learned theories to perform automatic annotation of new texts;
- visualizing and revising annotation results.

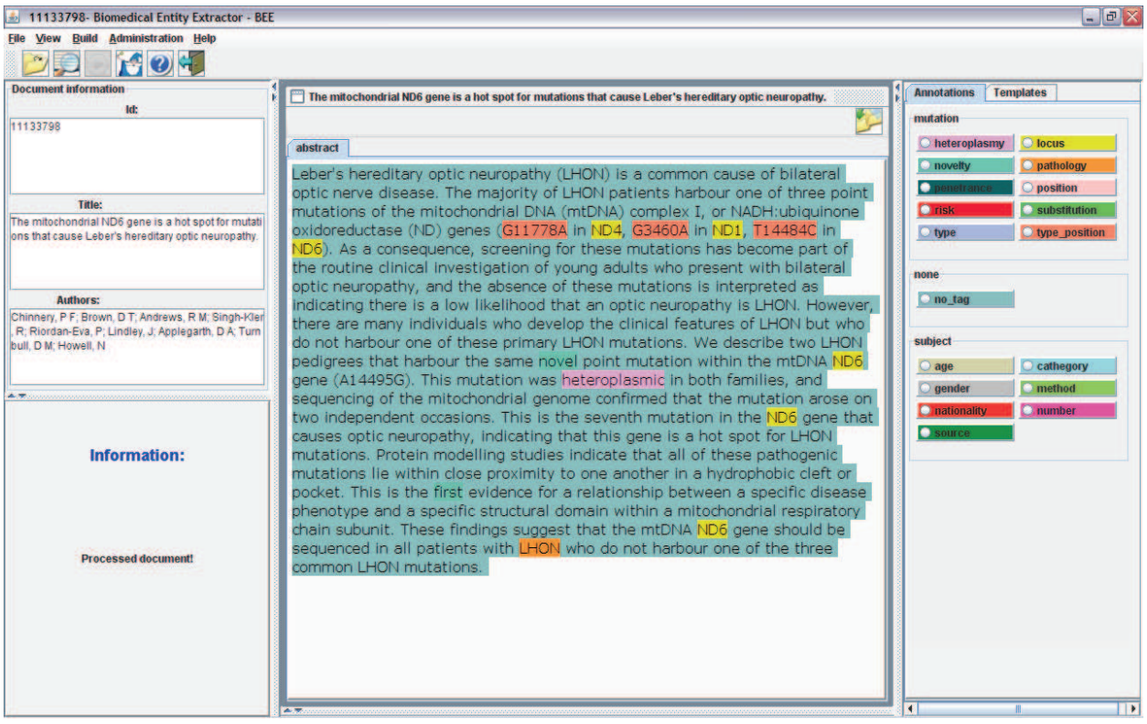



Fig. 1. BEE GUI

The BEE system includes a Graphical User Interface (GUI) which provides the user with facilities to customize the system for the specific information extraction problem. In

[4] http://www.di.uniba.it/~malerba/software/BEE

particular, domain dictionaries are submitted by selecting flat files and assigning a lookup name to each dictionary. Annotation schema are manually defined by grouping user-defined categories of named entities into templates. The GUI includes a wizard which supports the user in managing training sessions, i.e., data selection, choice of concepts to be learned, definition of learning parameters, specification of background knowledge, and running and monitoring learning tasks. Finally, the GUI allows users to manually associate tokens with categories on the basis of text pre-processing results, as shown in Figure 1.

The general architecture of the system is shown in Figure 2. The System Manager works by allowing user interaction and by coordinating the activity of all other components. It interfaces the system with the data persistence layer to store (1) information on texts concerning pre-processing results, feature extraction, associated annotations; (2) linguistic resources (i.e., gazetteers, acronym dictionaries, grammars); (3) annotation schemas of the biomedical problem at hand; (4) learned theories. The User Manager supports operations aiming to customize the system on the specific user-defined biomedical problem. The Text Processor is in charge of data elaboration and mapping into the learning descriptions operations described in Section 5.2 and 5.3. The output of this module allows users to invoke both the Recognition Module and the Learning Module through the GUI. The former is responsible for clause application and automatic association of annotation slots on text, the latter performs all the activities necessary to support learning sessions. Actually, the Recognition Module is able to match body parts of clauses available in the learned knowledge base with descriptions of new texts.

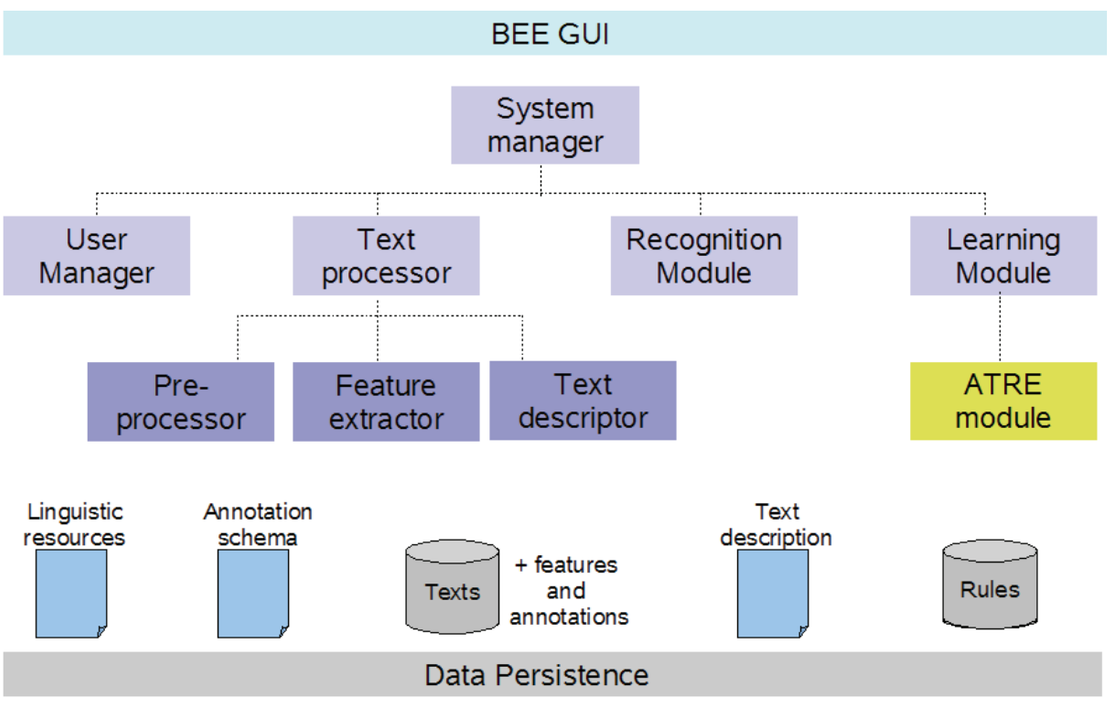



Fig. 2. BEE architecture

BEE is a java standalone application since it is conceived as an integrated environment for information extraction from texts, where curators define the annotation problem, prepare data, revise results, and learning experts manage learning operations. A web service version including the text processor and the recognition module is also available for collaborative environments. This web service is separately trained on application domains and made available together with knowledge bases.

## 6. Experiments

Fine tuning of the system on the HmtDB case study has been carried out within activities concerning the LIBI[5] (International Laboratory for BioInformatics) project. This projects aims at designing and setting up an advanced IT platform to support a newly-conceived Bioinformatics and Computational Biology laboratory "without walls". This includes tools enabling the deployment and maintenance of genomic, proteomic and transcriptomic databases, as well as the design and execution of new algorithms, and software for the analysis of genomes and their expression products. A collaborative environment has also been developed to boost both knowledge and resource sharing: researchers can share both data analysis tools, in the form of simple or composed (workflow) services and data, which are accessed through data federation mechanisms that allow their dislocation and heterogeneity to be bypassed. Available analysis tools cover not only typical bioinformatics algorithms supporting *in silico* molecular data handling and analyses, but also a suite of general-purpose text and data mining algorithms that enhance analysis capabilities of biological data managed by means of the federated database. Such an environment, where mining tools can benefit from the aggregated view of a plethora of different information sources provided by the federated database, is an ideal candidate where prototyping and testing systems devoted to semi-automated database annotation. To accomplish such a challenging task, data curation is one of the preliminary key steps. To this end, HmtDB has been federated together with other specialized data sources (including PubMed) and interfaced by the BEE miner to support mitochondrial genome curation activities.

We conducted an experiment on 130 full papers concerning mitochondrial mutations carefully selected for the annotation of HmtDB. In this phase, experiments were conducted on the mutation template, where benefits of the proposed learning method can be observed. Conversely, most of the issues of annotating subjects information can be almost fully satisfied by using regular expressions. Entities with a very low distribution of examples (i.e., risk, penetrance, novelty, heteroplasmy) were not considered in the experiment reported in this chapter. From the set of relevant papers, we obtained 368 target sentences out of 1040 sentences. Considering the total number of tokens used to describe sentences, the number of annotated tokens was 890, that is, 2.42 tokens per target sentence and 6.86 per paper, namely about 20.5% of the total number of tokens considered in the experiment. The remaining tokens, i.e., 3461, were considered as non-tagged (i.e., as negative examples for all concepts to be learned). Trainers have tagged a single occurrence of target concepts in the papers by preferring occurrences reported in the neighbourhood of other target concepts to be learned in order to discover intra-sentence dependence.

[5] http://www.libi.it

Performances are evaluated by means of a 5-fold cross-validation, that is, the set of 130 papers is firstly divided into five folds (see Table 3), and then, for every fold, ATRE is trained on the remaining folds and tested on the hold-out fold.
Results were evaluated according to several criteria. For each concept, we computed both the number of omission and commission errors and the value of precision and recall. Omission errors occur when annotations of tokens are missed, while commission errors occur when wrong annotations are "recommended" by some rule. The omission measure is reported as the ratio of the number of omission errors and the number of positive examples, and the commission measure as the ratio of the number of commission errors and the total number of examples. The recall measure is computed as the ratio of positive examples correctly annotated (i.e., true positives) and the sum of true positives and false negatives (i.e., omission errors). The precision measure is computed as the ratio of true positives and the sum of true positives and false positives (i.e., commission errors). The F-measure is the weighted harmonic mean of precision and recall, that is:

$$F - measure = \frac{precision \cdot recall}{precision + recall}$$

Experimental results are reported in Table 4 for each fold, while Table 5 reports accuracy values for each class.

| Fold | #sentence | #locus | #position | #substitution | #type | #type_position | #pathology | #no_tag |
|---|---|---|---|---|---|---|---|---|
| 1 | 71 | 37 | 12 | 5 | 8 | 31 | 69 | 650 |
| 2 | 76 | 39 | 8 | 6 | 5 | 56 | 73 | 735 |
| 3 | 75 | 49 | 6 | 6 | 7 | 57 | 83 | 712 |
| 4 | 70 | 35 | 7 | 6 | 10 | 39 | 52 | 633 |
| 5 | 76 | 42 | 15 | 8 | 13 | 39 | 67 | 731 |
| **Total** | 368 | 202 | 48 | 31 | 43 | 222 | 344 | 3461 |

Table 3. Distribution of examples per folds

| Fold | #locus | | #position | | #substitution | | #type | | #type_position | | #pathology | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **om** | **com** | **om** | **com** | **om** | **com** | **om** | **com** | **om** | **com** | **om** | **com** |
| 1 | 10.81 | 0.26 | 41.66 | 0 | 60 | 0.25 | 0 | 0.25 | 19.35 | 0.13 | 43.48 | 3.63 |
| 2 | 23.08 | 0.23 | 62.5 | 0.11 | 66.67 | 0 | 0 | 0 | 8.93 | 0.46 | 43.83 | 3.06 |
| 3 | 16.33 | 0.11 | 66.67 | 0 | 50 | 0 | 0 | 0 | 10.53 | 1.16 | 50.6 | 2.15 |
| 4 | 11.43 | 0.13 | 28.57 | 0 | 16.57 | 0 | 0 | 0 | 41.03 | 0.27 | 55.77 | 3.7 |
| 5 | 9.52 | 0.11 | 66.67 | 0.11 | 50 | 0 | 7.69 | 0 | 30.77 | 0.46 | 44.78 | 2.12 |
| ***Avg.*** | *14.23* | *0.17* | *53.21* | *0.04* | *48.67* | *0.05* | *1.54* | *0.05* | *22.12* | *0.5* | *47.69* | *2.93* |
| ***St.D.*** | *5.58* | *0.07* | *17.24* | *0.06* | *19.24* | *0.11* | *3.44* | *011* | *13.68* | *0.4* | *5.36* | *0.77* |

Table 4. Experimental results (percentage values): Average number and standard deviation of omission errors over positive examples and commission errors over negative examples

| Category | **Precision** | | **Recall** | | **F-measure** | |
|---|---|---|---|---|---|---|
| | Avg | St. Dev. | Avg | St. Dev. | Avg | St. Dev. |
| *locus* | 95.9 | 1.84 | 85.43 | 5.7 | 90.30 | 3.72 |
| *position* | 91.67 | 11.79 | 46.79 | 17.24 | 60.93 | 16.44 |
| *substitution* | 90 | 22.36 | 51.33 | 19.24 | 63.74 | 18.14 |
| *type* | 96 | 8.94 | 98.46 | 3.44 | 96.98 | 4.84 |
| *type_position* | 90.13 | 4.88 | 77.3 | 13.73 | 82.59 | 8.17 |
| *pathology* | 60.37 | 9.23 | 52.06 | 5.13 | 55.72 | 6.09 |

Table 5. Experimental results (percentage values): Mean and standard deviation of Precision, Recall and F-measure (β=1)

| Fold | #locus | #position | #substitution | #type | #type_position | #pathology |
|---|---|---|---|---|---|---|
| 1 | 165/35 | 36/15 | 26/11 | 34/5 | 191/52 | 275/116 |
| 2 | 163/30 | 40/14 | 25/10 | 37/5 | 166/54 | 271/119 |
| 3 | 153/36 | 42/13 | 25/10 | 36/5 | 165/33 | 261/110 |
| 4 | 167/38 | 41/16 | 25/11 | 32/5 | 183/47 | 292/120 |
| 5 | 160/37 | 33/15 | 23/11 | 29/4 | 183/39 | 277/116 |
| ***Avg.*** | *4.62* | *2.65* | *2.35* | *7.01* | *4.07* | *2.37* |

Table 6. Complexity of learned theories: number of positive examples over number of covered clauses per concept and average values

Performance variability for some concepts (e.g., *position*, *substitution*, *pathology*) among folds is due to different degrees of data sparseness, such as heterogeneity of examples and low percentage of positive observations available. However, the percentage of commission errors is very low with respect to that of omission errors (the system misses annotations rather than suggesting wrong ones) independently of the fold. This means that learned clauses are quite specific. By considering the complexity of learned theories (see Table 6), coverage rate can explain recall values. The best performances are obtained on the *type* class whose examples are the most homogeneous. Conversely, the worst performances are related to the annotation of a *pathology*. Actually, learning tasks for the *pathology* class appear to be intrinsically more complex, since we observe the highest percentage of commission errors despite the highest percentage of positive examples available. As regards the percentage of omission errors, we note that, while this is positively correlated to the number of discovered clauses, it is not correlated to the number of positive examples. This confirms the complexity of this annotation task. Low recall values and overfitted theories reflect difficulties mentioned in Section 4.4 concerning the variety of morpho-syntactic variations on the same pathology name, which leads to heterogeneous representations of examples. By scanning the learned theories, we observe that, for some classes, namely *substitution* and *position*, many clauses do take into account only lexical information specified by the predicate *word_to_string*. Indeed, on these entities the system performs the highest number of omissions and very few commissions. Concerning the *locus* and *type_position* entities, some omission errors were performed, in fact, good values of coverage rate are reported for theories learned for these concepts. By observing learned clauses, we found several clauses depending on lexical information but also some more general clauses as the followings:

```
annotation(X1)=locus ←follows_string_nn(X2)=mutation,
    word_category(X1)=gene, tfollows(X2,X1)=true

annotation(X1)=type_position ← char_number_char(X1)=true

annotation(X1)=type_position ← tfollows_string_nn(X2)=trnaser,
    type_of(X1)=alphanumeric
```

The first clause states that X1 is labelled as *locus* if it belongs to the gene category and it occurs in the sentence after the word "mutation". The second clause states that X1 is labelled as *type_position* if it is an alphanumeric token composed by a char, a number and another char. This is one of the first clauses that ATRE adds to the learned theory and covers many examples. Actually, information on type and position of a mutation is tokens such as A1262G, which means that A is substituted by G at position 1262 of the DNA. The third clause concerns the same concept and states that X1 is labelled as *type_position* if it is an alphanumeric token which is followed by the string "trnaser". This matches patterns where type and position information occurs in the neighbourhood of gene names (e.g., trnaser).
Clauses stating dependencies between these two concepts have been also discovered:

```
annotation(X1)=type_position ← annotation(X2)=locus,
    type_of(X1)=alphanumeric,
    distance_word_category(X2,X1)in[1.0..1.0]
```

It states that X1 is labelled as *type_position* if it is an alphanumeric string at distance one from a token labelled as *locus*.
Results show that annotation of concepts suffering from name mention ambiguity depends on the efficacy of the text pre-processing module in conjunction with the ability to exploit specialized lexical resources. Previous experiments, which are described in (Berardi & Malerba, 2007) where the usefulness of recursive theories is investigated, lead to different results. In particular, for concepts such as *type, locus* and *type_position* we got opposite findings since learned theories were very specific and constrained to lexical information. The new gene name dictionaries and the revised method for text tokenization and lexical patterns identification adopted to run experiments described in this chapter are able to keep under control morpho-syntactic variability of terms belonging to these classes.
Other meaningful clauses discovered in this experiment follow:

```
annotation(X1)=position ← annotation(X2)=substitution,
    tfollows_cd_nn(X1,X2)=true
```

This clause states that X1 is annotated as *position* if it is a numeric token that precedes a noun which has been annotated as *substitution.*

```
annotation(X1)=pathology ← follows_string_vb(X2)='trna(asn)',
    tfollows(X2,X1)=true
```

This clause states that X1 is annotated as *pathology* if it follows a verb preceded by the token 'trna(asn)', which is the name of a mitochondrial gene.

```
annotation(X1)=substitution ← first(X1)=a, last(X1)=g
```

This clause states that X1 is annotated as *substitution* if it is a token starting with the 'a' and ending with the 'g' characters, that are two nucleotide symbols. This is a peculiar clause

which allows to recognize all the mutations where the *A* base is substituted by the *G* base in a genome.

## 7. Conclusion

The maintenance of biological databases is currently a problem of great interest because the progress made in many experimental procedures has led to an ever increasing amount of data, mostly buried in textual form. In this chapter, we present a framework for biomedical information extraction from text that integrates a data mining module for extraction rule discovery. Patterns for biomedical entity extraction are induced from a set of manually labelled texts that are relevant for the application at hand. The mining process can exploit domain knowledge and search for dependencies among entities of interest. Application of the approach to the HmtDB annotation case study is described. Results show complexity of some learning tasks and usefulness of automatic text mining strategies. The mining system allows us to discover meaningful patterns among biomedical entities which can subsume some semantic relations, such as the association of a DNA mutation with the responsible gene. We are currently working to extend the framework by integrating a text classification system to automatically perform selection of literature that is relevant for the annotation task, which is an additional time-consuming and tiring task for curators. Since the work confirms that mining annotation rules offers a promising alternative to hand-coding, we plan to investigate approaches which are able to learn accurate models in the case of weakly labelled training data. This can alleviate the cost of producing complete training data which is a main drawback of supervised approaches. Moreover, weakly labelled data can be easily produced by exploiting the huge amount of knowledge already available in biological databases and by coupling it accurately with references that are provided as evidence of stored entries (Craven & Kumlien, 1999).

## 8. Acknowledgments

This work partially fulfills the research objectives set by the F.I.R.B. 2003 project LIBI (International Labratory of BioInformatics) funded by the MIUR (Italian Ministry for Education, University and Research) under grant RBLA039M7M (http://www.libi.it).

## 9. References

Accetturo, M., Santamaria, M., Lascaro, D., Rubino, F., Achilli A., Torroni, A., Tommaseo-Ponzetta, M., Attimonelli, M. (2006). Human mtDNA site-specific variability values can act as haplogroup markers. HUMAN MUTATION. vol. 27(9), pp. 965-974 ISSN: 1059-7794. doi:10.1002/humu.20365.

Aitken, J. S. (2002). Learning Information Extraction Rules: An Inductive Logic Programming approach. In F. van Harmelen (Ed.): Proceedings of the 15th European Conference on Artificial Intelligence, pages 355-359, IOS Press, Amsterdam.

Alfarano, C., Andrade, C.E., Anthony, K., Bahroos, N., Bajec, M., Bantoft, K., Betel, D., Bobechko, B., Boutilier, K., Burgess, E., et al. (2005) The Biomolecular Interaction

Network Database and related tools 2005 update. Nucleic Acids Res. 2005 Jan 1;33(Database issue):D418-24.

Andrews, R. M., Kubacka, I., Chinnery, P. F., Lightowlers, R. N., Turnbull, D. M., Howell, N. (1999) Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA. Nat Genet 23:147

Attimonelli, M., Accetturo, M., Santamaria, M., Lascaro, D., Scioscia, G., Pappada, G., Russo L., Zanchetta L. & Tommaseo-Ponzetta, M. (2005): Hmtdb, a human mitochondrial genomic resource based on variability studies supporting population genetics and biomedical research. BMC Bioinformatics 1(6) Suppl 4:S4.

Baumgartner, Jr. W. A., Cohen, K. B., Fox, L., Acquaah-Mensah, G., Hunter, L., (2007). Manual annotation is not sufficient for curating genomic databases. Bioinformatics 23:i41-i48.

Berardi, M. , Varlaro, A., Malerba, D. (2004). On the effect of caching in recursive theory learning. In Rui Camacho, Ross D. King, and Ashwin Srinivasan, editors, 14th International Conference on Inductive Logic Programming, ILP 2004, volume 3194 of Lecture Notes in Computer Science, pages 44–62. Springer.

Berardi, M., Malerba, D. (2007): Learning Recursive Patterns for Biomedical Information Extraction. In S. Muggleton, R. Otero, & A. Tamaddoni-Nezhad (Eds.): Inductive Logic Programming: ILP 2006, LNAI 4455, pages 79–93, Springer: Berlin.

Blockeel, H., Sebag, M., (2003). Scalability and efficiency in multi-relational data mining. SIGKDD Explorations, 5(1): 17-30.

Cohen, A.M. & Hersh, W.A. (2005). A survey of current work in biomedical text mining, Brief. Bioinform., 6(1):57-71.

Craven, M., Kumlien, J. (1999): Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pp. 77–86. AAAI Press, Stanford.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002) Gate: A framework and graphical development environment for robust nlp tools and application. In: Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia, USA.

Cussens, J., Nedellec, C. (ed.) (2005): Genic Interaction Extraction Challenge. Proceedings of the 4th ICML Workshop on Learning Language in Logic (LLL05), Bonn, Germany.

Gaizauskas, R. and Wilks, Y. (1998). Information Extraction: Beyond Document Retrieval. Computational Linguistics and Chinese Language Processing 3, 17-60.

Goadrich, M. , Oliphant, L., Shavlik, J. (2004). Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction. In Rui Camacho, Ross D. King, and Ashwin Srinivasan, editors, 14th International Conference on Inductive Logic Programming, ILP 2004, volume 3194 of Lecture Notes in Computer Science, pages 98-115, Springer.

Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A. (2005) . Overview of biocreative: critical assessment of information extraction for biology. Bioinformatics, 6, 2005.

Horn, F., Lau, A. L., Cohen, F. E. (2004) .Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. Bioinformatics 2004 20 (4): 557-568.

Horner, DS, Pesole, G. (2003) The estimation of relative site variability among aligned homologous protein sequences. Bioinformatics 2003, 19:600-606.

Jensen, L. J., Saric, J., Bork, P. (2006). Literature mining for the biologist: from information retrieval to biological discovery. Nature Reviews Genetics, Vol. 7, No. 2., pp. 119-129.

Krallinger, M., Erhardt, R.A., Valencia A. (2005) Text-mining approaches in molecular biology and biomedicine. Drug Discovery Today. 2005 Mar 15; 10(6):439-45.

Levi, G., and Sirovich, F. (1976) 'Generalized and-or graphs', Artificial Intelligence, 7, 243-259.

Lloyd, L. (1987). Foundations of Logic Programming, 2nd ed. Springer-Verlag.

Lu, Z., Cohen, K. B., Hunter, L. (2007). GeneRIF quality assurance as summary revision. Pac. Symp. on Biocomput., 12, 269–280.

Malerba D. (2003). Learning recursive theories in the normal ILP setting. Fundamenta Informaticae, 57(1):39–77.

Mitchell, T.M. (1997). Machine Learning. McGraw-Hill.

Muggleton, S. (1992). Inductive Logic Programming. Academic Press, London.

Nédellec, C. (2004). Machine Learning for Information Extraction in Genomics - State of the Art and Perspectives. In: Text Mining and its Applications: Results of the NEMIS Launch Conference Series: Studies in Fuzziness and Soft Computing Sirmakessis, Spiros (Ed.), Springer Verlag.

Nienhuys-Cheng, S.-W., de Wolf, R. (1997). Foundations of inductive logic programming. Springer, Heidelberg.

Pesole, G., Saccone, C. (2001). A novel method for estimating substitution rate variation among sites in a large dataset of homologous DNA sequences. Genetics 2001, 157:859-865.

Porter, M.F. (1997). Readings in information retrieval. An algorithm for suffix stripping, pp. 313–316

Ramakrishnan, G., Joshi, S., Balakrishnan, S., Srinivasan, A. (2007). Using ILP to Construct Features for Information Extraction from Semi-structured Text. In Hendrik Blockeel, Jan Ramon, Jude W. Shavlik, Prasad Tadepalli (Eds.): Inductive Logic Programming, 17th International Conference, ILP 2007, volume 4894 of Lecture Notes in Computer Science, pages 211-224, Springer 2008.

Rebholz-Schuhmann, D., Kirsch, H., Couto, F. (2005). Facts from Text—Is Text Mining Ready to Deliver?, PLoS Biology 3(2): e65

Shah, P.K., Bork, P., (2006). LSAT: learning about alternative transcripts in MEDLINE. Bioinformatics 22(7): 857-865

Shatkay, H., Feldman, R. (2003) Mining the biomedical literature in the genomic era: an overview. Journal of Computational Biology 10, 821–855.

Smeitink, J., van den Heuvel, L. & DiMauro, S. (2001) The genetics and Pathology of Oxidative phosphorylation. Nature Reviews Genetics 2001, 2:342-352.

Srinivasan, P. (2004). Text Mining: Generating Hypotheses from Medline. Journal of the American Society for Information Science, 55 (4), pp. 396-413.

Torroni, A., Rengo, C., Guida, V., Cruciani, F., Sellitto, D., Coppa, A., Calderon, FL., Simionati, B., Valle, G., Richards, M., Macaulay, V., Scozzari, R.: Do the four clades of the mtDNA haplogroup L2 evolve at different rates? Am J Hum Genet 2001, 69:1348-1356.

Wallace, D. C., Brown, M. D. & Lott, M. T. (1999) Mitochondrial DNA variation in human evolution and disease. Gene 1999, 238:211-230.

Yeh, A. S., Hirschman, L., Morgan, A. A. (2003). Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. Bioinformatics, Vol. 19 Suppl. 1

IntechOpen




**Data Mining in Medical and Biological Research**

Edited by Eugenia G. Giannopoulou

ISBN 978-953-7619-30-5
Hard cover, 320 pages
**Publisher** InTech
**Published online** 01, November, 2008
**Published in print edition** November, 2008

This book intends to bring together the most recent advances and applications of data mining research in the promising areas of medicine and biology from around the world. It consists of seventeen chapters, twelve related to medical research and five focused on the biological domain, which describe interesting applications, motivating progress and worthwhile results. We hope that the readers will benefit from this book and consider it as an excellent way to keep pace with the vast and diverse advances of new research efforts.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Margherita Berardi, Donato Malerba, Roberta Piredda, Marcella Attimonelli, Gaetano Scioscia and Pietro Leo (2008). Biomedical Literature Mining for Biological Databases Annotation, Data Mining in Medical and Biological Research, Eugenia G. Giannopoulou (Ed.), ISBN: 978-953-7619-30-5, InTech, Available from: http://www.intechopen.com/books/data_mining_in_medical_and_biological_research/biomedical_literature_mining_for_biological_databases_annotation

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen