A Lexicon for Biology and Bioinformatics: The BOOTStrep Experience

Valeria Quochi, Monica Monachini, Riccardo Del Gratta, Nicoletta Calzolari

Istituto di Linguistica Computazionale, CNR
Via Moruzzi 1, 56126 Pisa, Italy
E-mail: name.surname@ilc.cnr.it

Abstract

This paper describes the design, implementation and population of a lexical resource for biology and bioinformatics (the BioLexicon) developed within an ongoing European project. The aim of this project is text-based knowledge harvesting for support to information extraction and text mining in the biomedical domain. The BioLexicon is a large-scale lexical-terminological resource encoding different information types in one single integrated resource. In the design of the resource we follow the ISO/DIS 24613 "Lexical Mark-up Framework" standard, which ensures reusability of the information encoded and easy exchange of both data and architecture. The design of the resource also takes into account the needs of our text mining partners who automatically extract syntactic and semantic information from texts and feed it to the lexicon. The present contribution first describes in detail the model of the BioLexicon along its three main layers: morphology, syntax and semantics; then, it briefly describes the database implementation of the model and the population strategy followed within the project, together with an example. The BioLexicon database in fact comes equipped with automatic uploading procedures based on a common exchange XML format, which guarantees that the lexicon can be properly populated with data coming from different sources.

1. Introduction

As demonstrated, among other indices, by the increasing number of PubMed abstracts, Bio-literature is continuously being produced and new knowledge developed. It is of paramount importance to share and disseminate such knowledge in the biomedical domain especially for boosting and supporting discoveries of new treatments, medicaments, therapies.

The reuse of information however requires time and efforts because it usually involves integrating redundant and partial pieces of information, which are often stored in different formats. Consequently, intensive research work is being carried out to develop language technologies that provide intelligent access to such knowledge and build lexical and ontological resources to fulfill special demands for the biologist community.

Moreover, available bio-terminologies generally lack information relevant to knowledge extraction such as predicate argument structures and syntactic complementation patterns. Computational lexicons, on their turn, would hold the necessary detail of information, but do lack domain terminology and, above all, the linking to bio-ontologies, typical repositories of the kind of formal and conceptual information needed by knowledge capture systems.

It is a shared belief among the biomedical community that a comprehensive and continuously growing resource that integrates bio-terms from different sources encoded according to accredited standards, enriched with relevant linguist description and linked to concepts in the ontology would thus significantly improve text analysis and knowledge capture systems (Hahn and Marko 2001).

In the present paper, we report on the lexical terminological resource developed within the BOOTStrep project focusing on the description of the standard-based lexical model and its physical DB implementation.

The BOOTStrep BioLexicon aims at being a state-of-the-art lexical resource that meets both bio-domain requirements and the most recent standards for lexical representation. It is an integrated resource semi-automatically populated with data collected from different available biomedical sources (e.g. UniProt/ Swiss-Prot, ChEBI, BioThesaurus, NCBI taxonomy) and is further integrated with morphological, syntactic and lexical semantic properties either extracted from texts and or from domain resources.

The paper is organized as follows: we briefly report on some related works from which we took inspiration of with which we confront. Section 2 presents a description of the conceptual and XML model of the resource into its three main layers used for the representation of the three main linguistic levels: morphology, syntax and semantics. Section 3 presents some commented sample entries that show how the syntactic and semantic argument structure of predicative items is represented.

Section 4 describes the database implementation, the automatic uploading strategy, and presents some statistics of the current state of the database[1].

2. Related Works

As mentioned above, efforts have been dedicated to merge terms from different databases into one thesaurus, possibly with a normalized nomenclature (Kors *et al.* 2005) and to build extensible databases for storing terminological information aggregated across available sources, e.g. Termino (Harkema *et al.* 2004), lexical and ontological resources like the SPECIALIST lexicon (Browne and Srinivasan 2000). Still, access and interoperability of biological databases is hampered, due

[1] At the moment of writing, the BioLexicon DB is still missing complete syntactic and semantic information on the complementation patterns of predicative items. Extraction processes by project partners are still ongoing, albeit in its final stage. For this reason, the model for syntactic and semantic representation is in principle still subject to minor revisions. In the poster we will present updated figures, including syntactic and semantic information.

to persistent lack of uniformity of formats.

Concerning the representation of syntactic and semantic preferences of lexical items, subcategorisation patterns (or verb/predicate complementation, or valency) widely conceived are of key importance for various applications. Such information is used for example in processes for the automation of ontology engineering (see Hindle 1990, Pereira *et al.*1993, Faure and Nedellec 1998 among others for examples of the use of verb-object relations). A more recent example is the FP5 Dot.Kom project, where complex syntactic verb-argument dependencies are used as formal contexts of terms in order to construct taxonomies via machine learning techniques (Cimiano *et al* 2004). Complementation and semantic argument structure is also used to classify biomedical terms (see Spasic *et al* 2003). Subcategorization and argument structure information is particularly needed for event and fact extraction especially in the biomedical domain, where fact databases are of invaluable use for experimental researchers.

3. The BioLexicon Model

The BioLexicon is a computational lexicon for biology, designed to be reusable and flexible in order to be used by different applications: e.g. information extraction and information retrieval. Since one of the main aims is to foster semantic interoperability in the biology community, the ISO Lexical Markup Framework (Francopoulo *et al.* 2006a) was chosen as the reference meta-model for the structure of the BioLexicon. The Lexical Markup Framework, together with linguistic *constants* used for lexical description – i.e. the Data Categories[2]– provides a common, shared representation of lexical objects that allows for the encoding of rich linguistic information.

The BioLexicon accounts for (English) terms related to the bio-domain and represent morphological, syntactic and lexical semantic properties of them. Among these terms, especially relevant here is the encoding of biologically relevant verbs and nominalized forms of verbs, i.e. verbs typically used in biomedical texts to refer to bio-events. For such lexical items a full explicit representation of their syntactic complementation and of their semantic argument structure will be represented. The Biolexicon thus encodes those linguistic pieces of information that domain ontologies partially lack and which are, instead, important for information and knowledge extraction purposes.

Another key property and an innovation of the BioLexicon is that the Data Base comes equipped with automatic loading procedures for its population where data comes from project partners. Finally, term entries in the BioLexicon it will be linked to a BioOntology (a resource developed in parallel within the project) and both will serve as the terminological backbone for harvesting information from documents.

The BioLexicon is modeled in an XML DTD according to the Lexical Markup Framework (LMF) standard: it implements the core model plus objects taken from the NLP extensions for morphology, syntax and lexical semantics. The BioLexicon model, therefore is made of a subset of the lexical modules and lexical classes of the LMF standard.

The BioLexicon model consists of a number of independent lexical objects (or classes) and a set of Data Categories (DCs), i.e. attribute-value pairs which represent the main building blocks of lexical representation, especially tuned on the needs of the project. In conformity to the ISO philosophy, the Data Category Selection for the BioLexicon is partially drawn from the ISO 12620 Data Category Registry (ISO-12620 2006, Wright 2004, Ide and Romary 2004), and partially defined for the specific purposes of the project and the special domain. Furthermore, in order to be able to automatically constrain and check the consistency of the DCs on each specific object most DCs have been typed.

3.1 Data Categories

Data Categories (DCs hereafter) are the linguistic *constants* that are used to describe the single instances of the lexical classes. Data Categories take the form feature structures, or attribute-value pairs.

In conformity to the ISO philosophy, the Data Category Selection for the BioLexicon is partially drawn from the ISO 12620 Data Category Registry (Francopoulo *et al.* 2006), and partially integrated by defining a set of specific DCs needed for the representation of the domain terminology, whenever missing from standard repositories. In order to be able to automatically constrain and check the consistency of the DCs on each specific object, in the BioLexicon, most DCs have been typed.

In the following paragraphs we briefly describe each object of the model and indicate the kinds of data categories used or to be used to adorn them.

3.2 The Core Model

The core lexical objects of the BioLexicon are: *LexicalEntry, Lemma,* and *Sense*.

The *LexicalEntry* class represents the abstract units of vocabulary at three levels of description: morphology, syntax and semantics. To ensure modularity and extendibility the three levels of description are accounted for in separate lexical objects, independently linked to the *LexicalEntry*, which functions as a bridge among the *Lemma*, its related *Sense*(s), and *SyntacticBehavior*(s) (see fig. 1 for the core model and fig. 3 for the syntax extension).

[2] In the latest revision of the LMF specifications Data Categories are referred to as Features and *feat* is the XML corresponding element. In this paper we will continue to call them Data Categories (DCs).

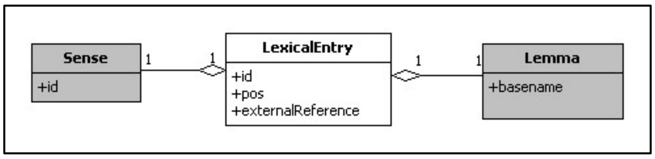

Fig. 1: The BioLexicon Core Model

LexicalEntry bears a Part-Of-Speech DC, plus additional attributes, such as the SourceDC, used to keep track of the id of the same term in other relevant resources. A specific requirement coming from the biology community is in fact that the resource should keep track of the ids of the terms in other well-known reference databases and ontology. External references in the BioLexicon are thus represented as typed Data Categories that are pointed at by the *LexicalEntry* object.

Lemma is used to represent the base form of lexemes plus possible additional grammatical properties[3]. The *Lemma* object is in a one-to-one relation with the *LexicalEntry*, which means that homonyms (and polysemous items) in the BioLexicon are represented as separate entries.

Finally, the basic information units at the semantic level are senses. *Sense* is the class used for the representation of the lexical meanings of a word/term. Each *Sense* instance represents and describes one meaning of a given *Lexical Entry*, may contains information on the specific (sub-)domain to which the sense applies, and will contain a link to the Bio-ontology.

Sense is the class used for the representation of the lexical meanings of a word/term, and it is inspired by the SIMPLE Semantic Unit (Ruimy *et al.* 2003).

3.3 The Morphology Extension

In a terminological lexicon for biology a key requirement is the representation of term variants. Variants in fact are extremely frequent and common in the biology literature (Nenadic *et al.* 2004). Given that linguistic information is automatically extracted from texts, in the BioLexcon we distinguish only two types of variants: variants of form and variants of meaning (semantic variants). These two types of variants are represented with different mechanisms. The morphology extension has been implemented mainly to allow for a rich and extensible representation of variants of form (see the diagram in fig. 2).

The *FormRepresentation* object in LMF has the function of representing multiple orthographies: a fundamental DC for this object is *writtenform*, which represents the string identifying the form in question.

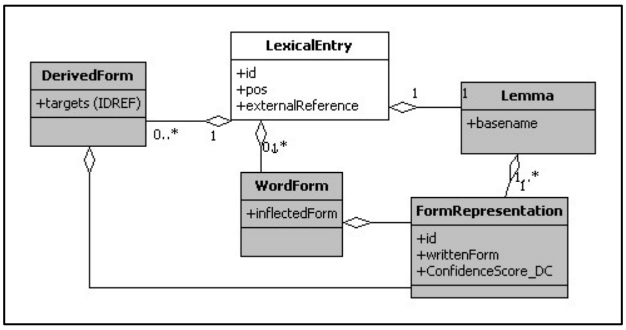

Fig. 2: The Morphology extension

In the BioLexicon we use this object to represent term graphic variants. Each variant is further adorned with special DCs, specifically devised to address the special needs of both the domain and the project. Therefore, a special DC is defined in order to account for confidence scores assigned to variants extracted from by means of machine learning techniques (see also Quochi *et al.* 2007).

The *WordForm* class in the BioLexicon is used to represent the automatically generated inflected forms of domain-relevant verbs.

3.4 The Syntax Extension

This section describes the module (i.e. the sets of classes) that allows for the representation of the syntactic combinatory properties of predicative lexical items through the set of objects related to *SyntacticBehaviour*.

The architecture of the syntax module and its lexical objects is designed taking into consideration the possible need to accommodate into the lexicon subcategorisation behaviors of terminological verbs automatically extracted from texts by appropriate NLP systems (and therefore the possibility of storing probability scores associated to each subcategorisation frame of a predicative item has been foreseen). The syntactic extension provides the structures for a detailed description of the syntactic behavior of a lexical entry. Fig. 3 shows a diagram of the syntax module.

SyntacticBehaviour is dedicated to the representation of how lexical items and terms are used in context. It represents one of the possible behaviors that a lexical entry shows in context by describing specific syntactic properties of a lexical item related to one of the possible contextual behaviors. *SyntacticBehaviour* is aggregated to *LexicalEntry* and optionally points to one or more senses (of the same *LexicalEntry*). It is therefore word/ term-specific.

The syntactic behavior of a lexical entry is moreover fully specified by the *Subcategorisation Frame*, the "*heart*" of the syntax module. The *Subcategorisation Frame* object is used to represent one syntactic configuration and does not depend on individual syntactic units; rather it may be shared by different units.

[3] Because all linguistic information encoded in the lexicon is to be automatically extracted from texts by project partners, for the moment there is no such grammatical information pertaining to the Lemma specifically (like gender for nouns). However, the picture of the population of verbs and verb-related information is not complete yet.

A *SubcategorisationFrame* describes the syntactic arity of a relational lexical unit, and through the *SyntacticArgument* class, is allows for a granular specification of their properties. The BioLexicon syntax extension additionally accommodates probability scores (i.e. the probability associated to each lexical item of appearing with a given complement configuration). Such probabilities are recorded in the form of a Data Category as a property of the *Syntactic Behavior* belonging to a given *SubcategorisationFrame*.

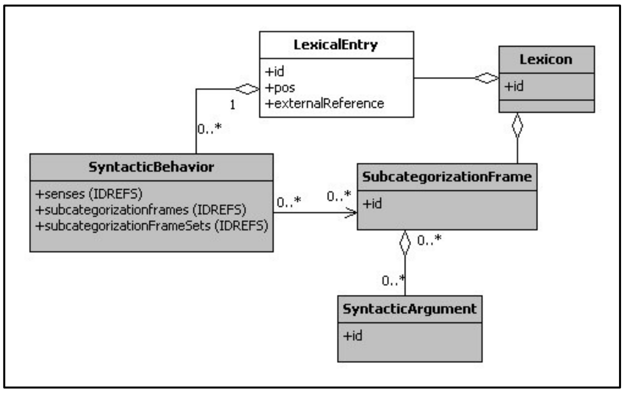

Fig. 3: The Syntax extension

3.5 The Semantic Extension.

The semantic module of the lexicon is made of lexical objects related to the *Sense* class. The representation of the semantic aspects of terms is entrusted in fact to the objects related and aggregated to *Sense* and *SemanticPredicate* (see Fig. 4).

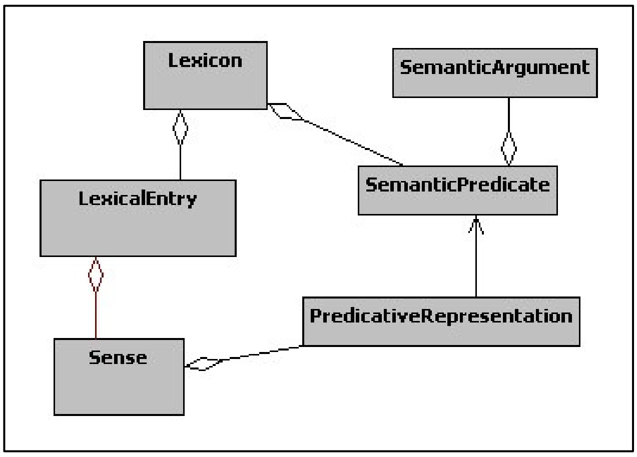

Fig. 4: The Semantic extension

Sense represents lexical items as lexical semantic units. Each Sense instance represents and describes one meaning of a given *LexicalEntry*, may contain information on the specific (sub)domain to which the term sense applies, and contains a link to the semantic type in the ontology which the sense instantiate.

Semantic relatedness among terms is an important property in the lexicon of natural languages and is used here also to account for semantic variants of terms. Semantic relatedness is expressed through the *SenseRelation* class, which encodes (lexical) semantic relationships among instances of the Sense class. BioLexicon semantic relations build on the 60 *Extended Qualia relations* of the SIMPLE model and are represented as Data Categories drawn from the Data Category Selection specifically defined to meet the needs of the bio-domain and of the BOOTStrep project (for details on bio-relations and the semantic extension in general see Monachini *et al*. 2007).

The *SemanticPredicate* class, instead, is independent from specific entries and represents an abstract predicative or relational meaning together with its associated semantic arguments. This meaning may be shared by more senses that are not necessarily considered as synonyms. In open domain lexicons it is typically shared by a verb and the corresponding nominalizations, so that it can link *LexicalEntries* that belong to different lexical classes. *SemanticPredicate* is referred to by the *PredicativeRepresentation* class, which represents the semantic behavior of lexical entries and senses in context, i.e. it describes the complete semantic argument structure of a predicative lexical item.

The *PredicativeRepresentation* class also encodes the type of link that a *Sense* holds with a *SemanticPredicate*, e.g. a verb, like *abolish*, which is a privileged realization of the predicate PREDAbolish, vs. its nominalization, *abolishment*.

Finally, the classes of the semantic module are the *loci* where the link between the BioLexicon and the conceptual resource of the project, the domain ontology, will be established.

3.6 Linking Syntax and Semantics

The mapping between syntactic and semantic arguments is realized via a mechanism inspired by the SIMPLE model (Ruimy *et al.* 2003). Subcategorization frames may be seen as representations of the surface realization of the semantic structure of predicates. Thus, by explicitly representing how semantic argument map onto syntactic ones, the lexicon is able to provide rich and useful information that can be used in the mining of texts to extract new facts and knowledge.

The *SynSemCorresp* and *SynSemArgMap* objects provide an explicit mapping of semantic arguments and roles onto syntactic slots, thus accounting for their surface realization (see the picture in Fig. 5).

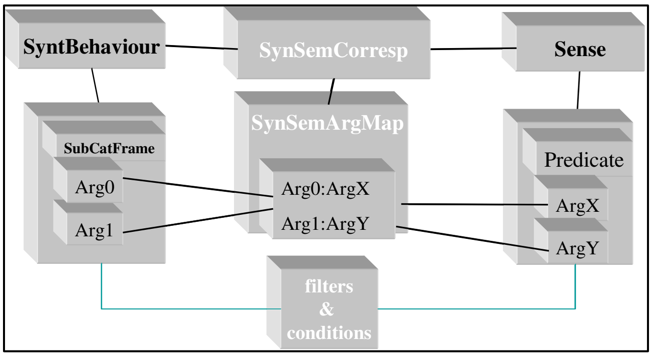

Fig. 5: Mapping the syntactic and semantic arguments

Let us take, for example, a regular subject-object transitive construction of the verb *abolish* (see Appendix): through the *SyntacticBehaviour,* the verb is linked to the *SemanticPredicate* PredAbolish, which, via *SemanticArguments,* specifies its argument structure. Syntactic arguments are specified as concerns their position (0, 1 …), function (subject, object …) and category (NP, PP …), while semantic arguments are defined in terms of their semantic roles (agent, patient) and semantic restrictions/ preferences of typical fillers (protein, gene …, i.e. semantic types in the ontology).

Now, the *SynSemCorrespondence* object is responsible for making explicit the mapping and specification of how each of the arguments of the two layers are mapped. Sample entries can be found in the Appendix. For the sake of simplicity, the example shows a very simple and prototypical case of isomorphic correspondence between syntax and semantics – a bivalent correspondence where arguments are in a one-to-one mapping. Obviously, in the lexicon other more complex types of correspondences will be defined.

4. The Automatic Population of the BioLexicon Database

The conceptual model has been implemented as a relational database capable of managing biological data both extracted from texts and collected from other existent resources. The BioLexicon DataBase (BLDB) consists of two modules: a MySQL database, and a java software component for the automatic population of the database. External to the BLDB, but fundamental for its automatic population, is an XML interchange format (XIF), which the java procedures parse and read to load data into the BLDB. The XIF thus allows for a standardization of the data extracted from the different terminological resources and from texts (by automatic NLP applications) and for the independency of both the uploading procedures and the BLDB from native data formats. The database is structured into three logically distinct layers:

1. the DICTIONARY FRAME contains tables used in the first handling of the XML Interchange Format and its rules that automatically build SQL instructions to populate target tables;
2. the STAGING FRAME is set of hybrid tables for volatile data;
3. the TARGET FRAME contains the actual BioLexicon tables i.e. tables that directly instantiate the BioLexicon DTD and contain the final data.

The neat separation between target tables (the BioLexicon proper) and “operational” tables allows for the optimization of the data uploading into the BLDB and ensures an easy extendibility both of the database and of the uploading procedures. In the near future, the database will be integrated in a UIMA framework and accessed either through APIs by software users or through a web graphic interface by various types of users with different needs. At present, the BLDB can be accessed and queried locally through a prototype graphic interface.

Currently, the BLDB contains terms and variants gathered by existing resources, with derived relations, and a set of automatically generated verb forms4. Table 1 and 2 give some statistics of the current state of the BLDB.

POS	Semantic Type	Lexical Entries	Variants
N	Enzyme	4.016	9.379
N	Gene/Protein	841.164	1.547.856
N	Species	367.565	439.336
N	Chemical	16.402	58.890
Total		1.229.147	2.055.461

POS	Semantic Type	Lexical Entries	Inflected Forms
V	--	591	2.941

Table 1: Number of LexicalEntries and of variants (of form) per semantic type and part-of-speech.

Relation Type	Instantiations
Is_a	483.937
Is_part_of	333
Is_synonym_of	628.409
Is_conjugate_base_of	905
Is_tautomer_of	248
Is_enantiomer_of	710
is_substituent_group_from	479
has_functional_parent	2644
is_conjugate_acid_of	905
has_parent_hydride	820

Table 2: Relation types and number of instantiations

4.1 A Practical Example

This section shows the three levels of the database “at work”. As explained before, input data are not directly loaded from the original resource, but is loaded from the XIF.

As shown in the XIF fragment below the Cluster element contains a set of coherent data encoded in specific sub-elements that represent linguistic notions. The Extraction Transformation Loading (ETL) process extracts (E) raw data from the input files[5], transforms (T) and loads it in temporal tables (staging tables) and finally

[4] Data are extracted and encoded in the XIF format by our EBI and NACTEM partners.

[5] Actually, data encoded in the XIF is not really “raw”, because, in order to produce them, some processing of the original DBs or raw texts was performed. However, from our point of view we can consider it as raw, since it needs further elaboration..

loads (L) it in the actual database tables (the target tables). The dictionary level of BLDB is logically divided into two separated parts: WORK and RULE.

The former manages the mapping of the XIF onto staging tables (E-T phases), while the latter deals with the upload of data into target tables (L-phase).

Staging tables have been modeled to be in a one-to-one correspondence with the XIF elements. Clearly also the element attributes are mapped to staging columns.

Let us consider the following example, from GeneProt:

```
<Cluster clsId="SC494014" SEMTYPE="GeneProt">
<Entry entryId="SC494014_1"
      baseForm="Isopullulanase precursor"
      type="PREFERRED">
   <SOURCEDC sourceName="UniProt"
sourceid="O00098"/>
   <POSDC posname="POS" pos="N"></POSDC>
   <Variant writtenForm="isopullulanase gene"
      type="orthographic"/>
   <DC att="swissprot_name" val="CISY_EMENI"/>
    <DC att="speciesNameNCBI" val="162425"/>
    </Entry>
</Cluster>
```

The WORK part of the Dictionary (WORK henceforth) maps XIF elements onto staging tables[6] (see Table 3 below).

XIF element	Staging table	Description
Entry	Lemma	This table contains lemmas
POSDC	LexicalEntry	This table contains lexical entry
SOURCE DC	LexicalEntry_ Source	This table contains all sources for a given lexical entry
Variant	FormRepresen tation	This table contains all different variants, acronym...

Table 3: Mapping between XIF elements onto Staging Tables.

Due to the design of the conceptual model, we decided to implement relations among objects as correspondence tables.

For instance, the Variant element in the XIF determines also the correspondence table between Lemma and FormRepresentation tables. This means that, while FormRepresentation contains a list of variants, the Lemma_FormRepresentation table contains variants defined for a given lemma. This is crucial since in the biological domain, the same orthographic form can be a variant of different lemmas.

Correspondence tables are defined both at staging and target level. Staging tables, therefore, contain raw data, which has to be subsequently manipulated in order to be loaded into target tables.

Let us consider, for instance, how the Variant element instantiates the FormRepresentation and the Lemma_FormRepresentation staging tables.

```
<Entry entryId="SC494014_1"
      baseForm="Isopullulanase precursor"
      type="PREFERRED">
   <Variant writtenForm="isopullulanase
         gene" type="orthographic"/>
</Entry>
```

WORK encodes information about the Entry and Variant elements. In details, it “knows” that the Variant element has its own identifier and that this identifier is built with a fixed rule. WORK also “knows” that the same element defines a correspondence table between itself and its parent element (Entry).

Let us show below how WORK creates input files for staging tables (for FormRepresentation and Lemma_FormRepresentation respectively):

```
“FR_isopullulanase gene”, “isopullulanase gene”, “orthographic”

“LM_Isopullulanase precursor”,“isopullulanase gene”
```

The direct benefit of using the dictionary level is that the loading software builds “objects”[7] on the basis of XIF elements contained at dictionary level and manages only these objects. This means that the mapping between XIF and staging tables is performed only once, during the E-T phase. Even the I/O operations are performed once per object as well as the loading of the data in the tables.

The second part of the dictionary is the RULE one (RULE hereafter). This part manages the mapping between staging and target tables and regulates the L-phase. This mapping is required since there is no one-to-one mapping between staging and target tables. RULE, therefore, maps source staging tables onto target tables and allows for the automatic creation of SQL instructions. These instructions are simply “SELECT..FROM...WHERE...” that, when executed, retrieve data from staging tables and save them in input files for target tables. We adopted this strategy to allow wide freedom in defining rules to populate target tables.

A typical example of L-phases is the decoding process that leads from the attribute -value pair to the corresponding identifier. Data categories, for example, are encoded in tables that are managed at L-phase of the loading process.

The staging FormRepresentation table contains the value “orthographic”, which identifies a type of variant. A data category VariantDC decodes this value in an identifier. RULE creates the following SQL instruction:

[6] In this example we describe the mapping between elements and tables only. The mapping between attributes and fields follows the same logic.

[7] These objects are managed as *javabeans*

```
"SELECT a.id,a.writtenform,d.id
  FROM FormrepResentation a, VariantDC b
  WHERE a.type=b.val".
```

When executed, this instruction produces the following input file ready to be loaded in the target FormRepresentation table:

```
"FR_ isopullulanase gene", "isopullulanase gene", "3"
```

RULE also creates objects on the target tables, which manage at once input files, SQL instructions and other features.

In conclusion, we can see the dictionary level as a middleware between the original data, encoded in XIF, and the actual database.

This structure of the database allows speeding up the loading process, since it is split into two different phases,: i) from XIF to staging tables and ii) from staging to target tables.

Just to add statistical information, all chemical data (more than 100,000 entries) are loaded in less than two minutes.

5. Final Remarks

This paper presented an application of the ISO standard Lexical Mark-up Framework to the design of a lexical-terminological resource that accounts for (English) terms related to the biology domain and will contains morphological, syntactic and lexical semantic properties of them. The paper focused especially on the encoding of syntactic and semantic properties of biologically relevant predicative items (verbs and nominalizations) that are used in the domain literature to refer to bio-events .

The architecture both of the model and of the DB is designed taking into consideration the need to accommodate into the lexicon also probabilistic information on the automatically extracted data (i.e. therefore the possibility of storing confidence scores for variants and probability scores for subcategorisation frames).

The representation of the semantic aspects of terms is entrusted instead to the objects related and aggregated to *Sense* and *SemanticPredicate*. Thus, by explicitly representing how semantic argument map onto syntactic ones, the lexicon is able to provide rich and useful information that can be used in the mining of texts to extract new facts and knowledge.

6. Acknowledgements

This work is carried out as part of the BOOTStrep project, an EU STREP project funded under the FP6 (FP6-028099).

We would also like to thank the three anonymous reviewers for their precious comments and suggestions.

7. References

Browne, McCray and Srinivasan 2000. *The Specialist Lexicon*. Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD.

Cimiano Philipp, Andreas Hotho and Steffen Staab. 2004. "Clustering Concept Hierarchies from Text" *Proceedings of the LREC 2004,* Lisbon, Portugal.

Faure D. and C. Nedellec. 1998. "A corpus-based conceptual clustering method for verb frames and ontology". In P. Velardi (ed.*) Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications.*

Francopoulo G. *et al.* 2006a. Lexical Markup Framework (LMF). *Proceedings of the LREC 2006*, CD-ROM, Genova, Italy.

Hahn U., Markó K. 2001. "Joint Knowledge Capture for Grammars and Ontologies". *Proceedings of the 1st international conference on Knowledge capture* Victoria, British Columbia, Canada.

Harkema H. *et al.* 2004. A Large Scale Teminology Resource for Biomedical Text Processing. HLT-NAACL 2004 *Workshop: Bio-LINK 2004*, Linking Biological Literature, Ontologies and Database, Boston, Massachusetts, USA.

Hindle. D. 1990. "Noun classification from predicate argument structures". In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Ide N. and Romary L. 2004. A registry of standard data categories for linguistic annotation. In *Proceeding of the LREC04,* Lisbon, Portugal.

ISO-12620. 2006. Terminology and other content language resources – Data Categories – Specifications of data categories and management of a Data Category Registry for language resources. ISO/TC37/SC3/WG4.

Kors J. A. *et al.* 2005. Combination of Genetic Databases for Improving Identification of Gens and Proteins in Text, Rotterdam, Netherlands

Monachini *et al.* 2007. "Lexical Relations and Domain Knowledge: The BioLexicon Meets the Qualia Structure". In *Proceedings of the GL2007 Conference*, 10-11 May 2007, Paris.

Nenadic, G., *et al.* 2004 Enhancing Automatic Term Recognition through Term Variation, in Proceedings of 20 th Int. Conference on Computational Linguistics, Coling 2004, Geneva, Switzerland.

Pereira F., N. Tishby and L. Lee. 1993. "Distributional clustering of English words". In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190.

Quochi V. *et al.* 2007. "Toward a Standard Lexical resource in the Bio Domain" In *Proceedings of 3rd L&TC*, 4-7/10/07 Poznań, 295-299.

Ruimy N., *et al.*. 2003. "A computational semantic lexicon of Italian: *SIMPLE*" In *Linguistica Computazionale*, Vol.XVIII-XIX: 821-864.

Spasic, I., Nenadic, G. and Ananiadou S. 2003 "Using Domain-Specific Verbs for Term Classification" In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 17-24.

Wright S.E. 2004. A global data category registry for interoperable language resources. *Proceedings of the LREC04*, Lisbon, Portugal.

Appendix: Sample XML Entries: the verb abolish and its nominalization abolishment

In this appendix we show two slightly simplified real-world entries in the BioLexicon, in order to show how complementation and predicate-argument patterns are represented and arguments mapped.

```
<Lexicon>
<!--###### Lexical Entries Start -->
<LexicalEntry id="LE_abolish">
<POSDC POSAtt="partOfSpeech"
POSVal="verb"></POSDC>
  <SyntacticBehaviour id="SB_abolish"
subcategorizationFrames="regularSVO"
senses="S_abolish">
  </SyntacticBehaviour>
  <Sense id="S_abolish">
<PredicativeRepresentation predicate="PredAbolish"
correspondence="bivalent">
        </PredicativeRepresentation>
  </Sense>
</LexicalEntry>
<LexicalEntry id="LE_abolition">
<POSDC POSAtt="partOfSpeech"
POSVal="noun"></POSDC>
  <SyntacticBehaviour id="SB_abolition"
subcategorizationFrames="PPofPPby"
senses="S_abolition">
  </SyntacticBehaviour>
  <Sense id="S_abolition">
<PredicativeRepresentation predicate="PredAbolish"
correspondence="CROSSEDbivalent">
        </PredicativeRepresentation>
  </Sense>
</LexicalEntry>
<!--###### Lexical Entries End -->
<!--###### Shared objects start -->
<!--###### SubcategorisationFrames -->
<SubcategorizationFrame id="regularSVO">
        <SyntacticArgument id="arg0regularSVO">
                <DC att="position" val="arg0"></DC>
                <DC att="function" val="subject"></DC>
                <DC att="syntacticConstituent"
val="NP"></DC>
        </SyntacticArgument>
        <SyntacticArgument id="arg1regularSVO">
                <DC att="position" val="arg1"></DC>
                <DC att="function" val="object"></DC>
<DC att="syntacticConstituent" val="NP"></DC>
        </SyntacticArgument>
</SubcategorizationFrame>
<SubcategorizationFrame id="PPofPPby">
        <SyntacticArgument id="arg0PPofPPby">
                <DC att="position" val="arg0"></DC>
                <DC att="function" val="object"></DC>
<DC att="syntacticConstituent"
val="PPof"></DC>
        </SyntacticArgument>
        <SyntacticArgument id="arg1PPofPPby">
                <DC att="position" val="arg1"></DC>
                <DC att="function" val="subject"></DC>
<DC att="syntacticConstituent" val="PPby"></DC>
        </SyntacticArgument>
</SubcategorizationFrame>
<!--###### SemanticPredicates -->
<SemanticPredicate id="PredAbolish">
        <SemanticArgument id="argXAbolish">
   <DC att="semFeature" val="argX"></DC>
          <DC att="role" val="Agent"></DC>
<DC att="restriction" val="Substance"></DC>
        </SemanticArgument>
        <SemanticArgument id="argYAbolish">
          <DC att="semFeature" val="argY"></DC>
          <DC att="role" val="Patient"></DC>
<DC att="restriction" val="Substance"></DC>
        </SemanticArgument>
</SemanticPredicate>
<!--###### Argument mappings -->
<SynSemCorrespondence id="bivalent">
<SynSemArgMap synFeature="arg0"
semFeature="argX">
</SynSemArgMap>
<SynSemArgMap synFeature="arg1"
semFeature="argY">
</SynSemArgMap>
</SynSemCorrespondence>
<SynSemCorrespondence id="CROSSEDbivalent">
<SynSemArgMap synFeature="arg0"
semFeature="argY"></SynSemArgMap>
<SynSemArgMap synFeature="arg1"
semFeature="argX"></SynSemArgMap>
</SynSemCorrespondence>
<!--###### Shared objects end -->
</Lexicon>
```

As it can be seen, a verb and the corresponding nominalization share the same predicate, and thus the same argument, but may be associated with different subcategorisation frames and may have different argument mappings. This way one accounts for both their different behavior and meaning and for their similarities.