A Framework for the Development of Biomedical Text Mining Software Tools

Anália Lourenço, Rafael Carreira, Sónia Carneiro, Paulo Maia, Daniel Glez-Peña, Florentino Fdez-Riverola, Eugénio C. Ferreira, Isabel Rocha and Miguel Rocha

***Abstract*—Over the last few years, a growing number of techniques has been successfully proposed to tackle diverse challenges in the Biomedical Text Mining (BioTM) arena. However, the set of available software tools to researchers has not grown in a similar way. This work makes a contribution to close this gap, proposing a framework to ease the development of user-friendly and interoperable applications in this field, based on a set of available modular components. These modules can be connected in diverse ways to create applications that fit distinct user roles. Also, developers of new algorithms have a framework that allows them to easily integrate their implementations with state-of-the-art BioTM software for related tasks.**

I. Introduction

Biomedical Text Mining (BioTM) [1] aims at the extraction of non-trivial information from biomedical documents. Traditionally, the act of literature curation was exclusively manual but the growing scientific publication rate, the continuous evolving of biological terminology and the complex analysis requirements brought by systems-level approaches urge for automated processes [2-4].

BioTM encompasses Information Retrieval (IR), Information Extraction (IE) and Hypothesis Generation (HG) as its main areas. IR deals with the automatic search and retrieval of relevant documents from the Web. According to pre-defined settings, retrieval programs (Web crawlers) visit and process an initial set of Web pages, looking for particular information and gathering hyperlinks for subsequent inspection, based on page relevance assessment.

Manuscript received July 5, 2008. This work was supported in part by the research projects recSysBio (ref. POCI/BIO/60139/2004) and MOBioPro (ref. POSC/EIA/59899/2004) of the University of Minho, financed by the Portuguese Fundação para a Ciência e Tecnologia. The work of SC is supported by a PhD grant from the same institution (ref. SFRH/BD/22863/2005).

A. Lourenço, S. Carneiro, E.C. Ferreira and I. Rocha are with the 1BB - Institute for Biotechnology and Bioengineering, Center of Biological Engineering – University of Minho, Campus de Gualtar, 4710-057 Braga – PORTUGAL (e-mails: {analia, soniacarneiro, ecferreira, irocha}@deb. uminho.pt).

R. Carreira, P. Maia and M. Rocha are with the Departament of Informatics / CCTC - University of Minho, Campus de Gualtar, 4710-057 Braga – PORTUGAL (e-mails: {rafaelcc, paulo.maia, mrocha} @di.uminho.pt).

D. Glez-Peña and F. Fdez-Riverola are with the Dept. Informática, University of Vigo, Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain (e-mails: {dgpena, riverola}@uvigo.es).

IE embraces all activities regarding automated document processing and combines Natural Language Processing (NLP) with Data Mining (DM) techniques. Stopword removal, tokenisation, stemming, shallow parsing and Part-Of-Speech (POS) tagging are commonly used in document pre-processing [5].

Named Entity Recognition (NER) deals with the identification of relevant biological entities (e.g. genes, proteins, compounds) [5-6]. Lookup tables, dictionaries and ontologies provide lexicon support to NER, but have shown to be insufficient due to the continuous evolving of biomedical nomenclature and term synonymy and homonymy issues (including term variants and abbreviations) [7]. Rule-based systems [6-8] delivered some automation by using templates (e.g. regular expressions) to describe term generation trends (e.g. the categorical nouns "ase" are commonly related to enzymes).

DM techniques are proposed to handle the mutating morphology and syntax of the terminology and to discriminate between ambiguous term senses. Techniques such as Hidden Markov Models (HMM) [9], Naive Bayes [10], Conditional Random Fields (CRFs) [11] and Support Vector Machines (SVMs) [12] have been applied to the annotation of controlled corpora. Those models are able to adapt to the constant evolving terminology while learning to detect specific entity patterns. Nevertheless, model prediction and generalization capabilities are highly dependent on the training corpus and the construction of a new corpus implies the laborious and time-consuming manual collection and annotation of a significant number (typically hundreds) of documents.

An increasing interest has been devoted to subsequent IE tasks. Relationship Extraction (RE), aiming at the identification of biologically relevant interactions and other events, represents the next logical step in terms of knowledge acquisition [13]. On the other hand, regarding HG, DM approaches have been applied to the conciliation of experimental data (e.g. omics data) with data retrieved from literature [9-11].

In spite of these achievements, BioTM strategies have focused so far on technique development rather than on cooperating with the biomedical research community and integrating techniques into workbench environments [12]. Freely available tools [13-17] fail to account for different user roles, presenting little flexibility or demanding expert programming skills. Furthermore, it is common that

information retrieval is constrained to abstracts, while there is important information in full texts.

Within this context, with the aim to address some of the above mentioned shortcomings, we present @Note, a novel framework for BioTM that invites both biologists and BioTM research teams to cooperate on application development. Its design directives were two-fold: (i) flexibility and interoperability aiming the inclusion and further extension of new BioTM approaches and (ii) transparency and simplicity, enabling the use of state-of-the-art techniques without requiring extensive previous knowledge about the undergoing activities.

@Note provides a reusable design for BioTM software systems and a set of pre-assembled software building blocks that programmers can use, extend and customize for their specific needs. It allows the development of BioTM software applications that integrate a number of NLP and DM tools and also support the major IR and IE processes.

The framework comprises core libraries and a set of reusable modules. The core libraries implement the general architecture and provide processing and visualization resources, including data representation, import and export features. The reusable modules overcome the need to recreate the same kind of resources or processes. On one hand, these modules handle basic language processing tasks such as POS tagging and lexicon-based semantic tagging. On the other hand, they support tasks such as feature selection, data mining and model evaluation. In fact, this component-based design allows for easy coupling of processes, facilitating the comparison of alternative configurations of the system (e.g. different dictionaries or relationship templates) or distinct implementations of the same module (e.g. different DM-based classification models).

II. IMPLEMENTATION

A. Overview

@Note is developed based on the idea that there are three different user perspectives for the whole platform: biologists, text miners and application developers (Fig. 1). Each role has specific needs that should be successfully addressed. For biologists, @Note can be seen as a set of user-friendly tools for biomedical text retrieval, annotation and curation.

Currently, a base application is made available that serves as an illustration of this role in the platform (to be further described in Section III). The available application covers the basic tasks involved in gathering and annotating scientific documents.

From a text miner perspective, the proposed system acts as a biological text analysis environment. It provides the user with a toolbox containing a set of techniques for text engineering, feature selection and classification that can be tuned, combined, executed and validated by the definition of custom experiments in a graphical manner.

Since @Note is intended to be a platform for research in the BioTM field, the role of developers is relevant and was also taken into account. It is expected that new services, algorithms or graphical components will be added frequently. Making changes, adding functionalities, integrating third-party software or new developments in the field should be performed as easy as possible. To achieve this goal, @Note was built on top of AIBench[1], a Java development framework, which increases the developer's productivity and where functionalities can be added and integrated in the form of plug-ins. These modules can be combined logically to achieve new applications and it is expected that the available set will steadily grow.

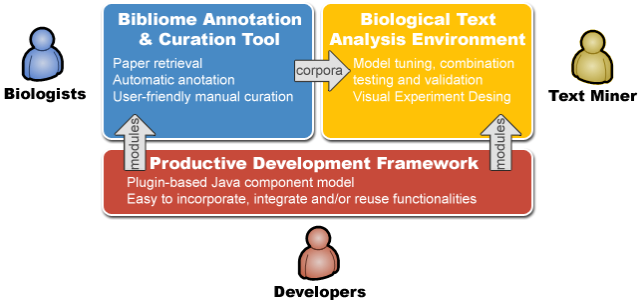

Figure 1- The distinct user roles contemplated by @Note

B. Functional modules

The framework integrates four main functional modules covering different tasks of BioTM (Fig. 2). Some parts of these modules were fully implemented from scratch, while others were developed using open-source software. The IR component is divided in the Document Retrieval Module (DRM) and the Document Conversion and Structuring Module (DCSM). The IE tasks are carried out by the Natural Language Processing Module (NLPM) and, finally, the Text Mining Module (TMM) deals with literature-specific DM tasks.

The DR module supports PubMed keyword-based query and document retrieval from open-access and subscribed web-accessible journals. When the user requests full text content, this module performs HTTP connections (Web crawling) to publishing journals (mentioned at PubMed entries). For open-access and subscribed Web-accessible journals, the document file (usually in PDF format) is automatically reached and downloaded. This functional module supports PubMed search via Entrez Programming Utilities (eUtils) [2] (implemented by the authors), while Perl LWP::Simple [3] and WWW::Mechanize[4] crawling modules were used in the development of the full-text journal retrieval module.

[1] http://www.aibench.org
[2] http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
[3] http://search.cpan.org/~gaas/libwww-perl-5.812/lib/LWP/Simple.pm
[4] http://search.cpan.org/dist/WWW-Mechanize/

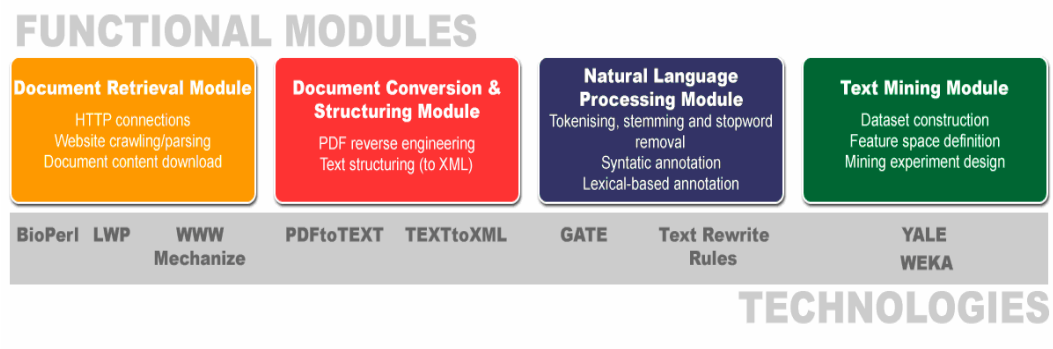

Figure 2 – Schematic representation of the @Note's functional modules.

The DCS module extracts text from the downloaded PDF documents. Once a document has been downloaded, the text is converted to plain text and then structured in XML format (title, authors, journal, abstract and the location of major sections are tagged). The conversion is carried out by the pdftotxt program, which is part of the Xpdf software[5]. The process of XML-oriented document structuring was implemented by the authors.

The NLP module embraces document pre-processing, lexicon-based NER and a friendly environment for manual annotation of documents. Documents (abstracts or full-texts) are submitted to tokenization, stemming and stopword removal processes, implemented using the GATE framework [19], which was integrated into @Note.

Following the pre-processing step, documents are annotated automatically, using a dictionary-based approach. Lexicon-based NER is sustained by a specialized Text Rewrite system developed by the authors upon Text::RewriteRules Perl module[6]. The system identifies document mentions to dictionary entries and applies templates to the general recognition of enzymes, RNA and DNA. Lexical resources can be created from major biomedical databases such as KEGG[7], BioCyc[8] or ChEBI[9] and integrated databases such as Biowarehouse[10].

The hierarchy of annotation classes was also created by the authors and aims at tracking down major biological entities. Currently, the system accounts for a total of 14 biological classes as follows: gene (including the subclasses metabolic and regulatory gene), protein (including the subclasses transcription factor and enzyme), pathway, reaction, compound, organism, DNA, RNA, physiological states and laboratory technique. All terms are normalized, i.e., synonyms and term variants are grouped around a concept label ("common" name) and the lemma of the verbs is extracted. At the best of our knowledge none of the existing NER tools covers such a large scope of biological classes nor offers different general and organism-specific dictionaries.

An additional novel feature of this work consists in the development of a manual annotation environment. This environment accounts for the review of automatic annotations by experts and the enhancement of the lexical resources. Also, the manually curated documents can be used as training corpus to build classification or other generalised learning models.

The TM module accounts for DM approaches to NER as well as to other analytical needs such as document relevance assessment or document clustering. It is implemented by low-level plug-ins to YALE[11] and WEKA[12] open-source DM toolkits, allowing the deployment of different problem-oriented mining experiments (feature selection and model evaluation).

Both the bibliographic catalogue and the lexical resources are kept in relational format (currently, on MySQL database engine). The workbench supports cooperative work by allowing server access to the database as well as to the documents. Users may benefit from previous queries, getting instant access to documents that have already been retrieved, but they may also cooperate on curation tasks, sharing locally processed documents.

A. *Implementation issues*

@Note supports continuous development, where new features and services can be added and improved frequently, integrating many research efforts. It is mainly developed using Java, due to the huge amount of freely available APIs and open source scientific developments, not to mention its other benefits such as object-orientation, language inter-operability, cross-platform nature, built-in support for multi-threading and networking, among others. @Note is built on top of AIBench[13] [20], a Java application development framework used in a growing number of research projects. This framework has three main advantages:

[5] http://www.foolabs.com/xpdf
[6] http://search.cpan.org/~ambs/Text-RewriteRules-0.11
[7] http://www.genome.jp/kegg/
[8] http://www.biocyc.org/
[9] http://www.ebi.ac.uk/chebi
[10] http://biowarehouse.ai.sri.com/

[11] http://rapid-i.com/]
[12] http://www.cs.waikato.ac.nz/ml/weka/
[13] http://www.aibench.org/

• It provides the programmer with a proven design and architecture. The applications developed with AIBench incorporate three types of well defined objects: operations, datatypes and datatype views, following the MVC (model-view-controller) design pattern. This leads to units of work with high coherence that can easily be combined and reused.

• It provides the programmer with services which are independent of the application scope, but useful for every application, like input dialog generation, application context management, concurrent operation execution, etc. The programmer can spend more time focusing in the problem specific requirements rather than in low level details.

• It is plug-in based. AIBench applications are developed adding components, called plug-ins, each one containing a set of AIBench objects. The coarse-grained integration between functionalities is carried out establishing dependencies between these plug-ins. This allows reusing and integrating functionalities of past and future developments based on AIBench.

Figure 3 shows the low-level integration perspective of @Note. @Note currently includes three main plug-ins: the core @Note plug-in, the GATE plug-in, and the YALE plug-in. The first one embraces the DR and the DCS functional modules, the second implements part of the NLP module and the third covers the TM functional module.

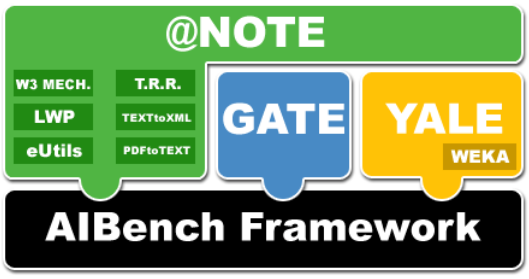

Figure 3 - Low-level integration perspective of @Note

The GATE and YALE plug-ins adapt these two well-known open source efforts in the area of Text Engineering and Data Mining respectively. They allow any AIBench application to include all the functionalities that these two environments provide. GATE and YALE were chosen because they are familiar to text and data miners, due to their open source nature and because they are also ongoing projects, where the new advances in their fields are rapidly included. Moreover, the YALE software also includes another popular data mining package, WEKA. The AIBench framework is also able to automatically generate technical documentation of the internal API of AIBench plug-ins (available in the web site), via another plug-in called Documentor.

B. Availability

The project is made available under the GPL license, together with documentation and other resources, in the project home page (http://sysbio.di.uminho.pt/anote.php).

III. SAMPLE APPLICATION

We exemplify the use of @Note with an application (for biologists) that implements a very common information flow: the retrieval of problem-related documents, their automatic annotation according to user pre-defined biological classes and user manual revision of document annotations (Fig. 4). This process is vital for many operations in biomedical research.

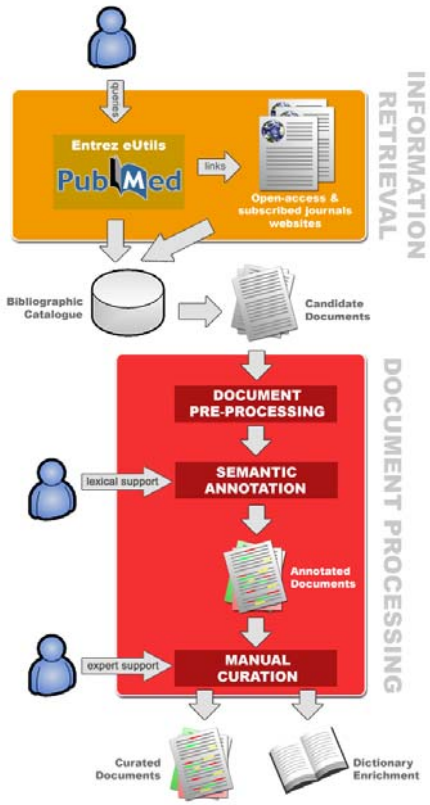

Figure 4 - Schematic illustration of a common literature retrieval and curation workflow

The search starts with a query issued to PubMed, and is followed by a traversal of available journals. After pre-processing, documents are automatically annotated using a dictionary selected by the biologist, who, in turn, will perform further revision of the annotations. Expert intervention will guarantee high-quality annotation and allow the enrichment of the dictionary.

Regarding IR, @Note is very simple to use: it is just a matter of introducing the set of words that better describe the problem under analysis. PubMed results will be crossed with previous query results and journal retrieval will only be deployed for documents that are not already available in the user's database.

After selecting the set of documents to be annotated (that can match or not the documents retrieved, since some IR results may be considered irrelevant), it is time to configure the automatic NER process. First, the user chooses the

dictionary that will sustain annotation. The dictionary can be selected from the set of dictionaries available at the server or created locally by the user from the current set of supported data sources. Then, he specifies which biological classes are to be considered, according to the selected dictionary and lookup tables.

The biologist revises dictionary-based annotation and when correcting or adding a new annotation, this previously unknown or mischaracterized information is reflected on the dictionary. Therefore, this process has two major outputs: high-quality annotation and dictionary enrichment. The latest is a classical example of a process of learning by experience that accounts well-known biological issues such as term novelty, term synonymy and term homonymy. Term novelty and the association of synonyms are far from being adequately tackled as they will depend on expert's knowledge, which is limited and often outdated just like dictionaries. However, the disambiguation of distinct mentions using the same term (e.g. same gene, protein and RNA name) is a classical example where manual curation is invaluable. Figure 5 depicts some of the features of the application.

IV. CONCUSION AND FURTHER WORK

The @Note project aims at fulfilling the existing gap between BioTM research and BioTM potential users. It was designed to target three different user roles: biologists, text miners and application developers. It provides user-friendly tools that aid users without BioTM expertise to manage and process the ever growing literature. Furthermore, it accounts for BioTM research needs, providing means to prepare and deploy NLP and DM experiments in well-known tools such as GATE, YALE and WEKA. It is built on top of AIBench framework, which facilitates the design and deployment of new applications as well as low-level tool integration. At the best of our knowledge, @Note is the first tool to integrate these three usage roles.

Another of its strengths is its integrated design that allows the development and evaluation of state-of-the-art BioTM techniques. The manual curation of automatic document annotation contributes to enhance lexicon support as well as to produce controlled corpora.

Given the nature of this project, the main effort in future work will be the development and integration of new functionalities, to be integrated as new @Note plug-ins.

ACKNOWLEDGMENT

We thank Alberto Simões and José João Almeida for helping deploy the text rewriting system and their expert suggestions in Natural Language Processing issues.

REFERENCES

[1] P. Zweigenbaum, D. mner-Fushman, H. Yu, and K. B. Cohen, "Frontiers of biomedical text mining: current progress," Briefings in Bioinformatics, vol. 8, no. 5, pp. 358-375, Sept.2007.
[2] S. Ananiadou, D. B. Kell, and J. I. Tsujii, "Text mining and its potential applications in systems biology," Trends Biotechnol., vol. 24, no. 12, pp. 571-579, Dec.2006.
[3] J. Natarajan, D. Berrar, C. J. Hack, and W. Dublitzky, "Knowledge discovery in biology and biotechnology texts: A review of techniques, evaluation strategies, and applications," Critical Reviews in Biotechnology, vol. 25, no. 1-2, pp. 31-52, 2005.
[4] K. B. Cohen and L. Hunter, "Natural language processing and systems biology," in Artificial intelligence methods and tools for systems biology. Dubitzky and Pereira, Ed. Springer Verlag, 2004.
[5] Fundel K, Zimmer R: Gene and protein nomenclature in public databases. BMC Bioinformatics 2006, 7.
[6] Liu HF, Hu ZZ, Torii M, Wu C, Friedman C: Quantitative assessment of dictionary-based protein named entity tagging. Journal of the American Medical Informatics Association 2006, 13: 497-507.
[7] Mukherjea S, Subramaniam LV, Chanda G, Sankararaman S, Kothari R, Batra V et al.: Enhancing a biomedical information extraction system with dictionary mining and context disambiguation. Ibm Journal of Research and Development 2004, 48: 693-701.
[8] Regev Y, Finkelstein-Landau M, and Feldman R, "Rule-based extraction of experimental evidence in the biomedical domain: the KDD Cup 2002 (task 1)," SIGKDD Explor. Newsl., vol. 4, no. 2, pp. 90-92, 2002.
[9] Yeganova L, Smith L, Wilbur WJ: Identification of related gene/protein names based on an HMM of name variations. Computational Biology and Chemistry 2004, 28: 97-107.
[10] Tsuruoka Y, Tsujii J: Improving the performance of dictionary-based approaches in protein name recognition. J Biomed Inform 2004, 37: 461-470.
[11] Sun CJ, Guan Y, Wang XL, Lin L: Biomedical named entities recognition using conditional random fields model. Fuzzy Systems and Knowledge Discovery, Proceedings 2006, 4223: 1279-1288.
[12] Dimililer N, Varoglu E: Recognizing biomedical named entities using SVMs: Improving recognition performance with a minimal set of features. Knowledge Discovery in Life Science Literature, Proceedings 2006, 3886: 53-67.
[13] A.Abi-Haidar, J.Kaur, A.Maguitman, P.Radivojac, A.Retchsteiner, K.Verspoor, Z.Wang, and L.M.Rocha, "Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks".," Genome Biology, 2008.
[14] Hoffmann R, Valencia A: Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics 2005, 21 Suppl 2: ii252-ii258.
[15] A. M. Cohen and W. R. Hersh, "A survey of current work in biomedical text mining," Brief. Bioinform., vol. 6, no. 1, pp. 57-71, Mar.2005.
[16] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text," Bioinformatics, vol. 21, no. 14, pp. 3191-3192, July2005.
[17] D. P. Corney, B. F. Buxton, W. B. Langdon, and D. T. Jones, "BioRAT: extracting biological information from full-length papers," Bioinformatics, vol. 20, no. 17, pp. 3206-3213, Nov.2004.
[18] Z. Z. Hu, I. Mani, V. Hermoso, H. Liu, and C. H. Wu, "iProLINK: an integrated protein resource for literature mining," Comput. Biol. Chem., vol. 28, no. 5-6, pp. 409-416, Dec.2004.
[19] Cunningham H: GATE, a general architecture for text engineering. Computers and the Humanities 2002, 36: 223-254.
[20] D. Glez-Peña, J.R. Méndez, P. Maia, M. Rocha; and F. Fdez-Riverola AIBench: a Java application framework for scientific software development. Science of Computer Programming. Submited for Publication. ISSN: 0167-6423. 2008.

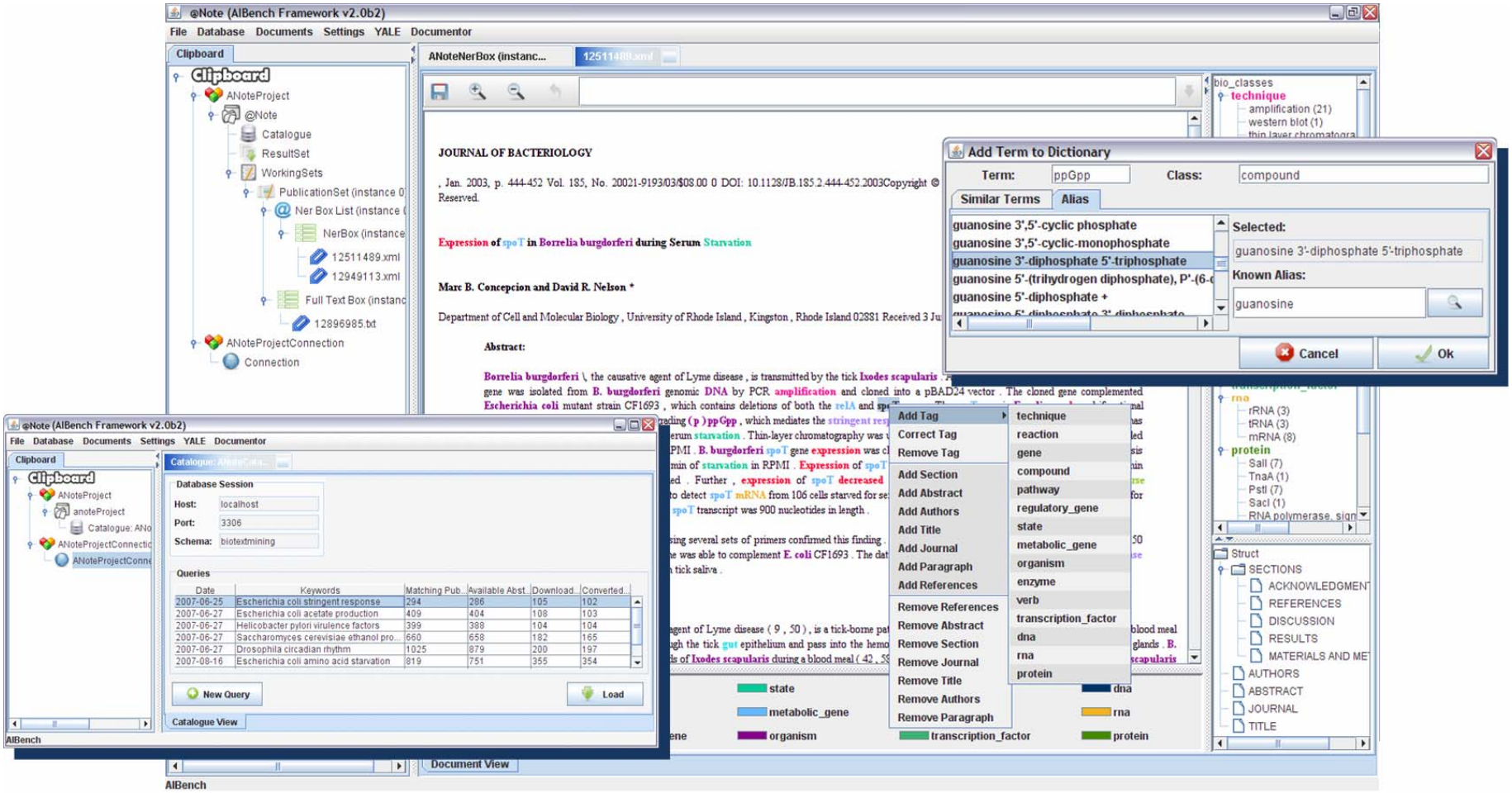

Figure 5 – Illustration of some of the features of the sample application.