Journal of Biomedical Informatics 42 (2009) 710–720

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

@Note: A workbench for Biomedical Text Mining

Anália Lourenço [a], Rafael Carreira [a,b], Sónia Carneiro [a], Paulo Maia [a,b], Daniel Glez-Peña [c], Florentino Fdez-Riverola [c], Eugénio C. Ferreira [a], Isabel Rocha [a], Miguel Rocha [b,*]

[a] IBB – Institute for Biotechnology and Bioengineering, Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
[b] Department of Informatics/CCTC, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
[c] Dept. Informática, University of Vigo, Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain

ARTICLE INFO

Article history:
Received 6 November 2008
Available online 22 April 2009

Keywords:
Biomedical Text Mining
Named Entity Recognition
Information Retrieval
Information Extraction
Literature curation
Semantic annotation
Component-based software development

ABSTRACT

Biomedical Text Mining (BioTM) is providing valuable approaches to the automated curation of scientific literature. However, most efforts have addressed the benchmarking of new algorithms rather than user operational needs. Bridging the gap between BioTM researchers and biologists' needs is crucial to solve real-world problems and promote further research.

We present @Note, a platform for BioTM that aims at the effective translation of the advances between three distinct classes of users: biologists, text miners and software developers. Its main functional contributions are the ability to process abstracts and full-texts; an information retrieval module enabling PubMed search and journal crawling; a pre-processing module with PDF-to-text conversion, tokenisation and stopword removal; a semantic annotation schema; a lexicon-based annotator; a user-friendly annotation view that allows to correct annotations and a Text Mining Module supporting dataset preparation and algorithm evaluation.

@Note improves the interoperability, modularity and flexibility when integrating in-home and open-source third-party components. Its component-based architecture allows the rapid development of new applications, emphasizing the principles of transparency and simplicity of use. Although it is still on-going, it has already allowed the development of applications that are currently being used.

1. Introduction

Nowadays, the ability to link structured biology-related database information to the essentially unstructured scientific literature and to extract additional information is invaluable for Computational Biology. Although an ever growing number of repositories is available, crucial theoretical and experimental information still resides in free text [1].

Biomedical Text Mining (BioTM) is a new research field [2] aiming at the extraction of novel, non-trivial information from the large amounts of biomedical related documents and its encoding into a computer-readable format. Traditionally, the act of literature curation, i.e. the inspection of a document and the extraction of relevant information, was exclusively manual. However, the outstanding scientific publication rate, the continuous evolution of the biological terminology and the ever more complex analysis requirements brought by systems-level approaches urge for automated curation processes [3–5].

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, taking advantage of available bibliographic catalogues and providing for local copies of potentially interesting publications whenever possible. IE embraces all activities regarding automated document processing, namely Named Entity Recognition (NER) [6–9] (also referred, along this work, as semantic tagging), Relationship Extraction (RE) [10–12], Document Classification [13,14], Document Summarisation (DS) [15,16] and the visualisation and traversal of literature data [17,18]. Its foremost aim is to emulate human curators, annotating biological entities of interest and relevant events (relationships between entities) in such a way that both document visualisation and further content analysis can deliver valuable knowledge. HG addresses the conciliation of literature-independent data (e.g. from laboratory or *in-silico* experiments) with the specific annotations derived from the literature, confirming IE results and assigning additional functional, cellular or molecular context [19–21]. In this paper, we will focus only on the IR and IE areas.

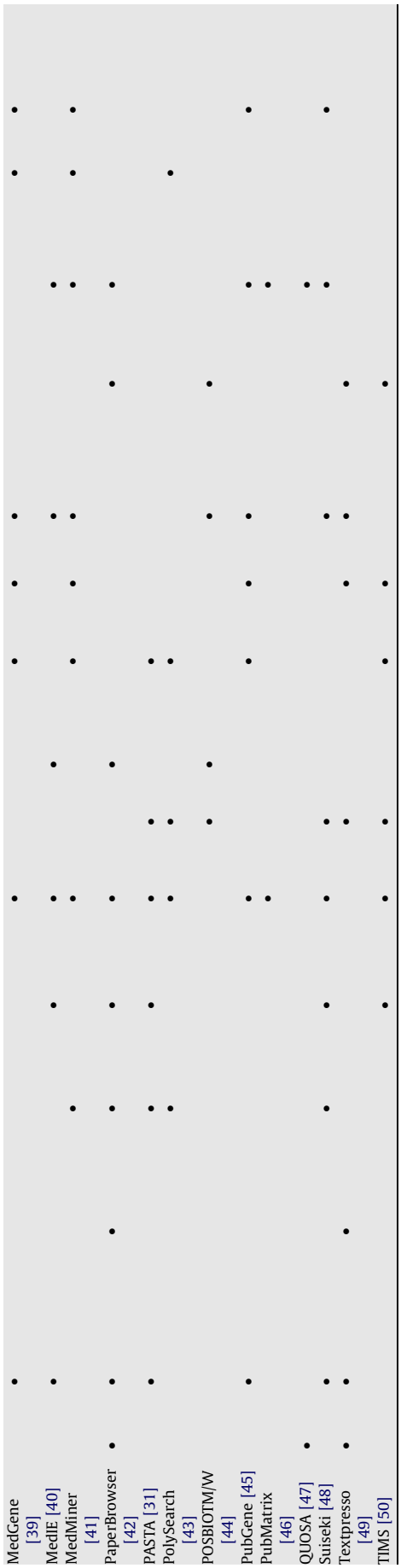
* Corresponding author.
E-mail addresses: analia@deb.uminho.pt (A. Lourenço), rafaelcc@di.uminho.pt (R. Carreira), soniacarneiro@deb.uminho.pt (S. Carneiro), paulo.maia@di.uminho.pt (P. Maia), dgpena@uvigo.es (D. Glez-Peña), riverola@uvigo.es (F. Fdez-Riverola), ecferreira@deb.uminho.pt (E.C. Ferreira), irocha@deb.uminho.pt (I. Rocha), mrocha@di.uminho.pt (M. Rocha).

1532-0464/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2009.04.002

Table 1
Feature comparison of several BioTM tools. There are numerous BioTM tools available. To compare them in terms of features can be somewhat difficult as many have emerged of particular goals (e.g. gather information about a certain organism or recognise all protein mentions) and therefore, their capabilities may be quite relevant for those goals but may seem limited at a general view. Our tool comparison got together the main contributions of each tool and, at the same time, identifies gaps or limitations within its scope of application.

	Full text	Organism/problem specific	Information retrieval			
			PubMed search	Other search engine	Journal crawling	Bibliographic catalogue
ABNER [27]						
AliBaba [28]			•			
BioIE [29]	•	•	•			
Chilibot [20]			•	•		
EBIMed [25]			•			
EDGAR [30]		•	•			
EMPathIE [31]		•				
GAPSCORE [32]		•				
GeneWays [33]	•	•	•		•	
GIS [34]		•	•			
GoPubMed [35]			•			
iHOP [36]		•	•	•		
LitMiner [37]			•	•		
MedEvi [38]			•			
MedGene [39]		•	•			•
MedIE [40]		•				
MedMiner [41]			•	•		
PaperBrowser [42]	•	•				
PASTA [31]		•				
PolySearch [43]			•	•		
POSBIOTM/W [44]			•			
PubGene [45]		•				•
PubMatrix [46]			•			•
QUOSA [47]	•		•		•	•
Suiseki [48]		•	•			
Textpresso [49]	•	•				
TIMS [50]						

	Full texts	Organism/problem specific	Information extraction												
			Pre-processing			Semantic tagging			Lexicon resource		Semi-automatic pos-processing				
			PDF->TXT conversion	Basic processing[a]	Syntactic tagging[b]	Lexicon[c]	Rules	Machine learning	Creation	Extend	Relationship extraction	Manual curation	Corpus navigation[d]	Link out[e]	Hypothesis generation
ABNER [27]				•				•							
AliBaba [28]				•	•	•			•		•		•	•	
BioIE [29]	•	•		•			•								
Chilibot [20]					•		•		•		•		•	•	•
EBIMed [25]				•		•			•		•		•	•	
EDGAR [30]		•			•	•	•		•		•				
EMPathIE [31]		•		•	•	•	•		•		•				
GAPSCORE [32]		•		•	•		•	•							
GeneWays [33]	•	•		•	•	•	•		•		•		•		
GIS [34]		•		•		•			•	•	•				
GoPubMed [35]				•		•							•		
iHOP [36]		•		•		•			•	•			•	•	
LitMiner [37]						•			•				•		
MedEvi [38]				•		•			•				•	•	



[a] Basic processing includes tokenisation, stemming, stopword removal and sentence delimitation.
[b] Syntactic tagging (Part-Of-Speech, shallow parsing).
[c] Dictionaries, ontologies and lookup tables.
[d] Context-rich relationship networks or simple links between terms and documents/document sentences.
[e] Link out to Web-accessible biological databases.

1.1. Main existing efforts in IR and IE

Acknowledging the existence of numerous efforts in IR and IE, it is important to establish which are the current achievements and limitations at the different tasks and, in particular, identify areas where contribution is most needed. A comparison of the main features of a set of selected available tools is given in Table 1.

Usually, Biomedical IR tools [18,22,23] exploit the search engine of PubMed [24], which is currently the largest biomedical bibliographic catalogue available. PubMed provides for general publication data (e.g. title, authors and journal) and, whenever possible, also delivers the abstract and external links for Web-accessible journals. Abstracts only provide for paper overview and thus, the retrieval of full-text documents is considered desirable for most applications. However, few tools support Web crawling into Web-accessible journals, limiting IE output to general knowledge acquisition.

There is a large diversity of tools that perform IE tasks, using alternative approaches. Document pre-processing and NER are key tasks in these tools. Document pre-processing involves document conversion, stopword removal, tokenisation, stemming, shallow parsing and Part-Of-Speech (POS) tagging (also referred as syntactic tagging), among other tasks [25].

The conversion of conventional publishing formats (e.g. PDF and HTML) into more suitable processing formats (namely plain ASCII) is prone to errors and information losses. Issues regarding the conversion of Greek letters, superscripts and subscripts, tables and figures are still open [26]. Also, conventional English shallow parsing and POS tagging do not comply with biological terminology and some efforts have been made to use benchmarking biomedical corpora in the construction of specialised parsers and taggers [27–29].

NER deals with the identification of mentions to biological entities of interest in biomedical texts. Strategies for NER combine Natural Language Processing (NLP) with Machine Learning (ML) techniques [30,31]. Lookup tables, dictionaries and ontologies provide first-level support [32–34] to NER. Rule-based systems [35–37] deliver additional automation by using templates (e.g. regular expressions) to describe well-known term generation trends in domain-specific problems (a classical example is the categorical nouns "ase" that are commonly related to enzyme mentions). ML techniques are used to create NER models, capable of encompassing the mutating morphology and syntax of the terminology and discriminating between ambiguous term senses. Techniques such as Hidden Markov Models (HMM) [38], Naive Bayes methods [39], Conditional Random Fields (CRFs) [9] and Support Vector Machines (SVMs) [40,41] have been successfully applied to the annotation of controlled corpora (e.g. Genia [42], BioCreAtIvE [34,43] or TREC [44,45]).

However, most NER tools focus on gene and protein tagging and the annotation of new biological classes demands major restructuring in terms of both annotation schema and resources. Also, it is difficult to find NER tools that enable the on-demand construction of lexical resources, i.e. dictionaries and ontologies.

ML-oriented approaches are typically based on benchmarking over particular corpora and constructed using a particular algorithm. Currently, no tool provides a user-friendly workflow for the construction of new models and model evaluation, i.e. feature selection and comparison between different algorithms.

Moreover, at this point, biomedical annotated corpora represent a bottleneck in the development of software as current approaches cannot be extended without the production of corpora, conveniently validated by domain experts. Computational tools for annotation already exist [46–48], but issues such as the support to semi-automatic annotation (using and creating resources such as dictionaries, ontologies, templates or user-specified rules), flex-

ibility in terms of annotation schemas and data exchange formats and the definition of user-friendly environments for manual annotation are usually not contemplated in such tools.

1.2. Motivation and aims

So far, most BioTM strategies have focused on technique development rather than on cooperating with the biomedical research community and integrating techniques into workbench environments [49]. Freely available tools (see Table 1 for references) fail to account for different usage roles, presenting little flexibility or demanding expert programming skills. This limits the application of new approaches to real-world scenarios and, consequently, the use of BioTM from the end-user perspective.

With the aim of providing a contribution to close this gap, we propose @Note, a novel BioTM platform that copes with major IR and IE tasks and promotes multi-disciplinary research. In fact, it aims to provide support to three different usage roles: biologists, text miners and application developers (Fig. 1).

For biologists, @Note can be seen as a set of user-friendly tools for biomedical document retrieval, annotation and curation. From a text miner perspective, it provides a biological text analysis workbench encompassing a number of techniques for text engineering and supporting the definition of custom experiments in a graphical manner. The developer role addresses the inclusion of new services, algorithms or graphical components, ensuring the integration of BioTM research efforts. Making changes, adding functionalities, integrating third-party software or new developments in the field can be performed in an easy manner.

@Note aims to provide support to each of these three roles individually, but also to sustain the collaborative work between users with different perspectives. In summary, @Note's primary aims by role are as follows:

- Allow the biologists to deal with literature annotation and curation tasks using a friendly graphical application.
- Allow the biologists to take advantage of novel text mining techniques, by the easy utilisation of ready-to-use models which can partially automate manual tasks like text annotation and relevant document retrieval.
- Allow the text miners to use and configure Bio-TM models without programming.
- Allow the text miners to translate to the biologists their configured and validated models in order to use them in real-world scenarios.
- Allow the developers to continuously provide or integrate new functionalities in modular applications.

The next section describes @Note's implementation, in terms of its design principles, of the high level functional components and also of its low-level development details. Each usage role is characterised in terms of operational needs and resources, identifying the support provided by @Note. Its usage in research groups that host researchers with distinct profiles is exemplified in Section 3, with a use case regarding the collection of data from the literature for a particular biological phenomenon, an example of a task to be performed by a biologist. Another example deals with the development and validation of ML models for NER (by text miners) and its subsequent use by biologists over their curated data. The two applications described in that section provide examples of @Note's potential use and illustrate its design principles.

2. Design and implementation

The three usage roles present in @Note stand for three expertise levels in terms of BioTM usage and programming. Biologists are not expected to have extensive knowledge about BioTM techniques or programming skills. Text miners are knowledgeable in BioTM techniques, but are not able to program the inclusion of new techniques or the adaptation of existing ones, focusing on the analysis of different BioTM scenarios. Developers are responsible for the programming needs of both biologists and text miners, adding or extending components and, eventually, including third-party components.

Thus, the design of @Note was driven by two major directives. Firstly, it provides developers with tools that aim at the inclusion and further extension of BioTM approaches, by considering the following development principles: (i) *modularity*, by promoting a component-based platform, both providing a set of reusable modules that can be used by developers to build applications and also supporting the possibility of developing and integrating new components; (ii) *flexibility*, by allowing the available components to be easily arranged and configured in diverse ways to create distinct applications; and (iii) *interoperability*, by allowing the integration

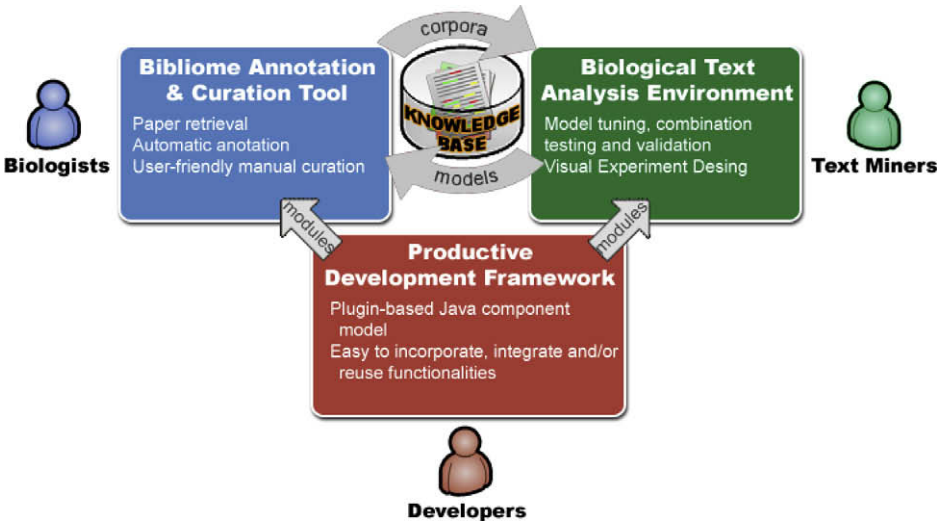


Fig. 1. The three distinct usage roles contemplated by the @Note workbench.

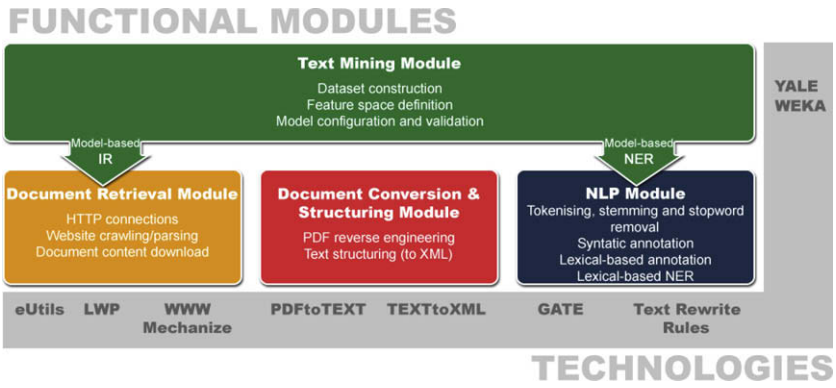


Fig. 2. Tasks and techniques at @Note's functional modules. A scheme showing the main tasks executed in each functional module.

of components from different open-source platforms that can work together into a single application.

Secondly, it seeks to provide the final users with applications developed under the principles of (i) *simplicity*, providing easy-to-use and intuitive user interfaces and (ii) *transparency*, enabling the use of state-of-the-art techniques without requiring extensive previous knowledge about the undergoing activities.

In the next subsections, the @Note workbench will be presented at two levels of abstraction. On one hand, we describe the *functional modules*, the technologies that were used to carry out their implementation and the available resources. In particular, we explain the inclusion of features from third-parties such as GATE text engineering framework [50] and YALE data mining workbench [51] in @Note's modules. On the other hand, we detail the *low-level integration* in terms of software modules, module integration and how the whole system can be extended.

2.1. Functional modules

@Note integrates four main functional modules covering different tasks of BioTM (Fig. 2). The Document Retrieval Module (DRM) accounts for IR tasks. Initial IE steps are covered by the Document Conversion and Structuring Module (DCSM), whereas the Natural Language Processing Module (NLPM) supports tokenisation, stemming, stopword removal, syntactic and semantic text processing. In particular, the SYntactic Processing sub-module (SYP) carries out POS tagging and shallow parsing, while the Lexicon-based NER sub-module (L-NER) and the Model-based NER sub-module (M-NER) are responsible for semantic NER annotation. Finally, the Text Mining Module (TMM) deals with ML algorithms, providing models for distinct IR or IE tasks (e.g. NER or document relevance assessment).

2.1.1. Document Retrieval Module

PubMed is currently the largest biomedical bibliographic catalogue available and it accepts external/batch access through the Entrez Programming Utilities (eUtils) Web service [52]. It provides trivial document metadata (such as title, authors and publishing journal) and, whenever this information is available, delivers the abstract, the MeSH keywords [53] and the links to Web-accessible journal sources.

Our DRM supports PubMed keyword-based queries, but also document retrieval from open-access and subscribed Web-accessible journals. It accounts for the need of processing full-text documents, in order to obtain detailed information about biological processes. The module exploits the eUtils service, following up its user requirements, namely ensuring a 3 second delay between requests. On the other hand, Perl LWP::Simple [54] and WWW::Mechanize [55] crawling modules were used in the development of the full-text retrieval functionality.

External links are traversed sequentially, avoiding server overload and respecting journal policy. The module identifies most document source hyperlinks through general templates. However, for journals where traverse is not straightforward (for example, due to javascript components or redirect actions), particular retrieval templates need to be implemented. Moreover, before issuing document retrieval, each candidate hyperlink is tested using the head primitive, ensuring that the document is retrievable and its MIME type corresponds to a PDF file. File contents are compared with the corresponding bibliographic registry in order to ensure that the document has been actually found.

Apart from implementing the search and retrieval of problem-related documents, the DRM also supports document relevance assessment. Keyword-based queries deliver a list of candidate documents and the user usually evaluates the actual relevance of each of these documents. Even taking into account document annotations, this process is laborious and time-consuming as some assessments demand careful reading of full-texts and the interpretation of implicit statements.

Foreseeing the need to automate relevance assessment, the module includes ML algorithms to obtain problem-specific document relevance classification models, thus delivering some degree of automation to this process.

2.1.2. Document Conversion and Structuring Module

The DCSM is responsible for PDF-to-text document conversion and first-level structuring. PDF files need to be translated to a format that can be utilised by posterior NLP modules. Plain ASCII text is considered the most suitable format, but this conversion implicates numerous information losses. Since current PDF-to-text processors are not aware of the typesetting of each journal, two-column text, footnotes, headers/footers and figures/tables captions (and contents) tend to be dispersed and mixed up during conversion. Also, there are terminology-related issues such as the conversion of Greek letters, superscripts and subscripts, hyphenisation and italics.

After testing several PDF conversion tools, including existing software for Optical Character Recognition (OCR), we concluded that no tool clearly outperformed the others and most of the aforementioned problems persisted. For now, @Note includes two of the most successful free conversion programs, namely: the pdftotext program (which is part of the Xpdf software [56]) and its MAC OS version [57] and the PDFBox [58].

The process of XML-oriented document structuring was based on bibliographic data and general rules. @Note catalogue provides for title, authors, journal and abstract data. Additional template rules search for known journal headings (such as Introduction,

Implementation, Conclusions and References), assuming that they are usually fully capitalised (or present an initial caps) and start at the beginning of a line and are followed by a newline.

2.1.3. Natural Language Processing Module

The NLPM embraces document pre-processing, syntactic annotation, semantic annotation and a friendly environment for the manual annotation of documents. Furthermore, it is able to process abstracts and full-texts interchangeably.

The interchange of annotation schemas, the management of terminological resources (namely dictionaries) and the use of existing syntactic annotators have been fully detailed in [59]. Here, we aim at providing a general overview on our work in this topics and, in particular, to establish the main path required to produce annotated documents.

Tokenisation, sentence splitting and stopword removal are the basic text processing steps, and typically they do not rely on previous pre-processing, whereas shallow parsing and NER may be based on POS annotation. In fact, the developed tools are able to deal with both semantic and syntactic annotation and annotation processes have no precedence over one another, i.e. semantic annotation may occur after or before POS tagging. Such multi-layer annotation may support text mining tasks (namely the construction of NER classifications models) as well as further relationship extraction.

Basic text processing steps were implemented using GATE features. Syntactic annotation is outputted by GATE's POS tagger whereas semantic annotation, i.e. NER, may be sustained by lexical resources (L-NER) or a classification model (M-NER). The L-NER sub-module was fully developed by the authors and incorporates a lexicon management plug-in and a specialised system to rewrite text using regular expression-based rules developed upon Text::RewriteRules Perl module [60].

This module also supports the construction and use of lexical resources, encompassing data loaders for major biomedical databases such as BioCyc [61], UniProt [62], ChEBI [63] and NCBI Taxonomy [64] and integrative databases such as Biowarehouse [65]. Also, it provides lists of standard laboratory techniques, general physiological states and verbs commonly related to biological events produced by the authors.

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 state and laboratory technique. The rewriting system attempts to match terms (up to 7-word composition) against dictionary contents, checking for different term variants (e.g. hyphen and apostrophe variants) and excluding too short terms (less than 3-character long). Additional patterns are included to account for previously unknown terms and term variants. For example, the template "$([a-z]\{3\}[A-Z]+\backslash d^*)$" (a sequence of three lower-case letters followed by an uppercase letter and a sequence of zero or more digits) is used to identify candidate gene names while the categorical nouns "ase" and "RNA" may point out to unknown enzyme and RNA entity mentions, respectively. Besides class identification, the system also sustains term normalisation, grouping all term variants around a "common name" for visualisation and statistical purposes.

The M-NER sub-module aims at applying classification models to the NER task and therefore accounting for the constantly mutating biological terminology. Text miners use the TMM (described next) to build such models. Nevertheless, no expertise on text mining is required to run the M-NER sub-module and thus, any user may use existing models and configurations on a particular problem (see the example on Section 3).

Both the L-NER and M-NER sub-modules provide invaluable aid to curators, but available techniques do not fully cope with terminological issues. Manual curation is still an important BioTM requirement and @Note acknowledges this fact by providing a user-friendly environment where biologists (problem experts) may revise automatically annotated documents. The manual annotation environment guarantees high-quality annotation and hence the extraction of relevant information. Annotated documents resulting from L-NER can be refined, eliminating or correcting (e.g. change term class or adjusting term grams) existing annotations and adding new annotations. Such annotation refinement may also support dictionary updates, accounting for term novelty and term synonymy.

Manually curated documents can be used as training corpus at the TMM to build classification models. In fact, the existence of this curation environment makes it possible for biologists and researchers to cooperate in the improvement of BioTM corpora to build automated models upon expert-revised knowledge.

2.1.4. Text Mining Module

The TMM accounts for the workbench for conducting text mining experiments. The module is implemented by a low-level plug-in to YALE [51], that also includes WEKA [66,67]. These are two open-source toolkits, allowing the deployment of different problem-oriented text mining experiments (namely feature selection and model evaluation).

Currently, the module aims only at the construction and evaluation of NER ML models that can be further used at the M-NER sub-module of the NLPM, although other tasks such as document relevance are already being developed. NER-oriented dataset preparation was implemented by the authors upon GATE features and covers morphological, syntactical and context features. Morphological features track term composition elements (such as capitalization, hyphenisation, alphanumeric data, quotes and tildes) and affix information (3–5 character long). Syntactical features are based on POS tagging. Context features capture the morphological and syntactical nature of the words in the neighbourhood of the term (typically, two words for each side).

Based on their expertise, text miners select the set of features that better describe each problem and perform mining experiments. Experiments evaluate different mining algorithms and alternative algorithm configuration. The resulting model can then be saved and further used in the M-NER sub-module.

2.1.5. Resource management

In @Note there are three main resources: the bibliographic catalogue, the lexical resources and the documents. The bibliographic catalogue is fed up by the DRM, storing PubMed record details (e.g. title, authors and journal), source links, log data on journal accessibility and document relevance assessments. The lexical resources include the dictionaries derived from biomedical databases and the lookup tables. Dictionaries are dynamically created by the user according to the particular annotation problem and the availability of database loaders. Regarding lookup tables, currently those are available for the annotation of physiological states, laboratorial techniques and biologically meaningful verbs.

Both the catalogue and the lexical resources are kept in a relational database (MySQL) that can be located at the user's local machine (private) or at a remote server (shared).

Documents retrieved for user-specified queries are kept locally along with the corresponding plain text, structured and annotated files. Thus, users get instant access to documents that have already been retrieved for prior queries and may use and compare results from different annotation procedures through the sharing of processed documents.

2.2. Low-level Integration Issues

At the low-level, @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, which has found increased adoption in the scientific community, 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 interoperability, cross-platform nature, built-in support for multi-threading and networking, among others.

@Note is built on top of AIBench [68], a Java application development framework used in a growing number of research projects. This framework has three main advantages:

- AIBench 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.
- AIBench 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.
- AIBench 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.

The use of AIBench makes an outstanding contribution in the pursuit for the declared design principles of @Note, namely:

- *Interoperability*: AIBench allows the developer to integrate under a MVC-based design different functionalities which can come from other third-party software. AIBench promotes the creation of datatypes and operations in order to wrap the proprietary structures of independent software into standardized formats, allowing the interoperability inside the final application.
- *Flexibility*: AIBench is a highly configurable and a field-independent framework which facilitates changes in a continuously developing application.
- *Modularity*: mainly using the plug-in engine (at a higher level) and the concepts of datatype and operation at a lower level.

The AIBench framework is also able to automatically generate technical documentation of the internal API of AIBench plug-ins (integrated in the Developer's manual available in the web site), via a plug-in called Documentor. This is a valuable item both for the developers involved in the @Note project and for developers interested in using the available plug-ins to develop new applications.

Currently, @Note includes three main plug-ins: the core @Note plug-in, the GATE plug-in, and the YALE plug-in (Fig. 3). The core plug-in encompasses modules fully developed by the authors (e.g. DRM, DCSM and L-NER) while the other two plug-ins adapt well-known open-source efforts in the area of Text Engineering and ML respectively. 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. By the adaptation of YALE to AIBench, the use of Weka algorithms is straightforward.

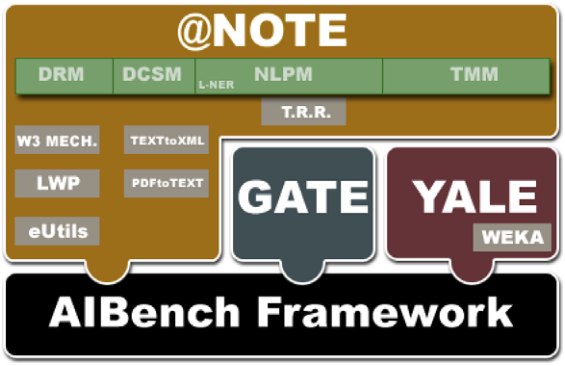


Fig. 3. Low-level integration perspective of @Note. AIBench comprises core libraries and delivers a set of functionalities in the form of plug-ins. Currently, AIBench integrates GATE text engineering plug-in and YALE data mining plug-in.

3. Results

We demonstrate the strengths of @Note in a real-world scenario embracing different usage roles. Imagine that a biologist or biotechnologist is interested in the study of stress responses triggered by nutrient starvation (i.e. stringent response) in *Escherichia coli*. This phenomenon is quite common when *E. coli* is used for the production of biopharmaceuticals and it reduces the productivities that can be obtained [69]. Therefore, its full characterization is indispensable for the identification of strategies to overcome it.

The ultimate goal of this study would then be to obtain a systems-level view of the biological process, by characterizing the mechanisms in the basis of this response. A major part of this characterization is the identification of the main entities involved (genes, proteins, metabolites, etc.), since they can be targets for strategies to inhibit this stress response. Although some data can be retrieved from public databases, the major part of the information still lies in the literature. Moreover, it is not likely that such information could be retrieved from abstracts, since detailed molecular mechanisms are usually described in the full text. A user-friendly tool like @Note could then be of major interest for cases like this, since it allows automatically retrieving and partly curating relevant documents. From the biologist perspective and for this problem, the workflow would set as follows (Fig. 4): search PubMed for available information and, whenever possible, retrieve full-text documents; automatically process the set of publications and, in particular, perform lexical-based NER with an organism-specific dictionary; manually revise some or all annotations, ensuring the reliability of further information extraction; and inspect the overall corpus results.

Implicitly, this process is incremental, as new documents are always appearing and their contents should be incorporated in the overall view. As such, the biologist is interested in refining not just document annotations, but also the lexical resources supporting NER which will enhance posterior knowledge acquisition stages. Furthermore, the biologist may consider using more advanced approaches for NER by posing his problem to a text miner that will create NER classification models. The text miner will be responsible for dataset preparation and the evaluation of different ML techniques. By comparing the performance of those techniques, he will propose a configuration for ML-based model for NER. The biologist will determine whether the model outperforms lexicon based NER results and, if not, may provide the miner additional information about the problem.

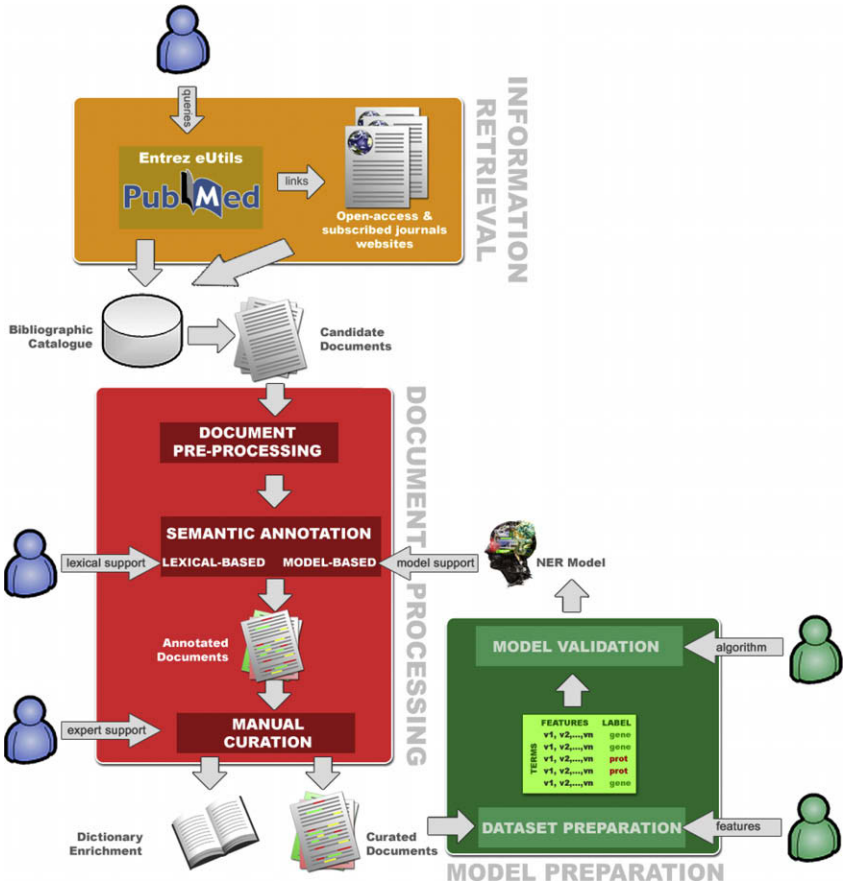


Fig. 4. Schematic illustration of a scenario where biologists and text miners cooperate.

3.1. The Biologist perspective – looking for documents on E. coli stringent response and delivering an annotated corpus

All operations defined within this section can be performed using the @Note Basics (+ML-based models) application provided in the project's web site. Furthermore, the detailed steps of this task and relevant screenshots are given in Supplementary Material (also available at the web site). At first, the biologist defines the keyword-based query as "*Escherichia coli* stringent response". Automatically, @Note deploys a PubMed search and information on the documents (including abstracts) is retrieved (Fig. 5). As we are interested in full-text contents, pdf document retrieval is also issued, but only for documents that are not already catalogued (obtained in previous queries). Process duration will depend on the

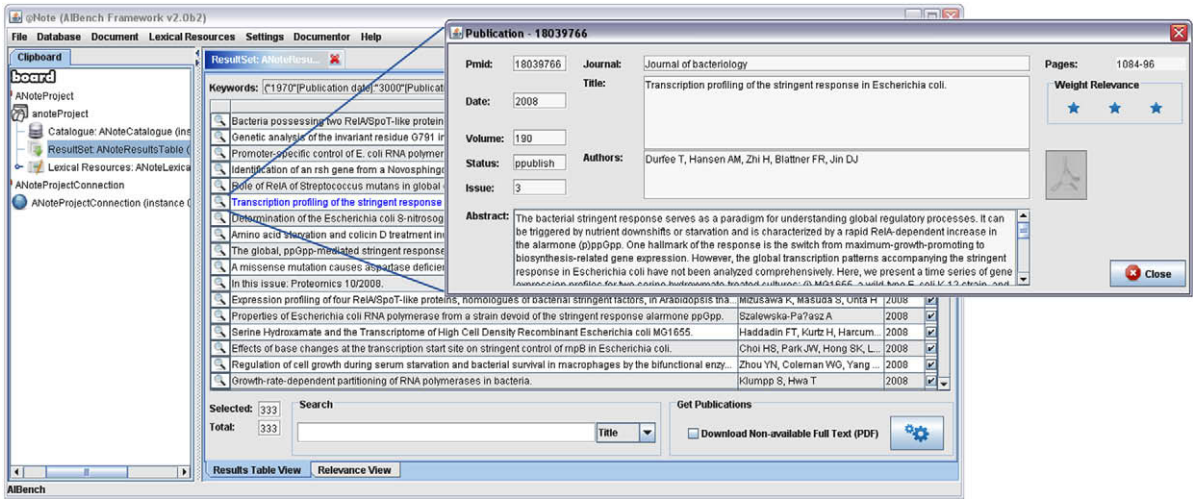


Fig. 5. Information retrieval with @Note: obtaining and visualising documents from PubMed related to a specific query. The biologist asks PubMed for publications related to a specific topic and, in order to get detailed information, journal crawling is also issued. @Note stores the outputted documents into the bibliographic catalogue, supporting further access and processing.

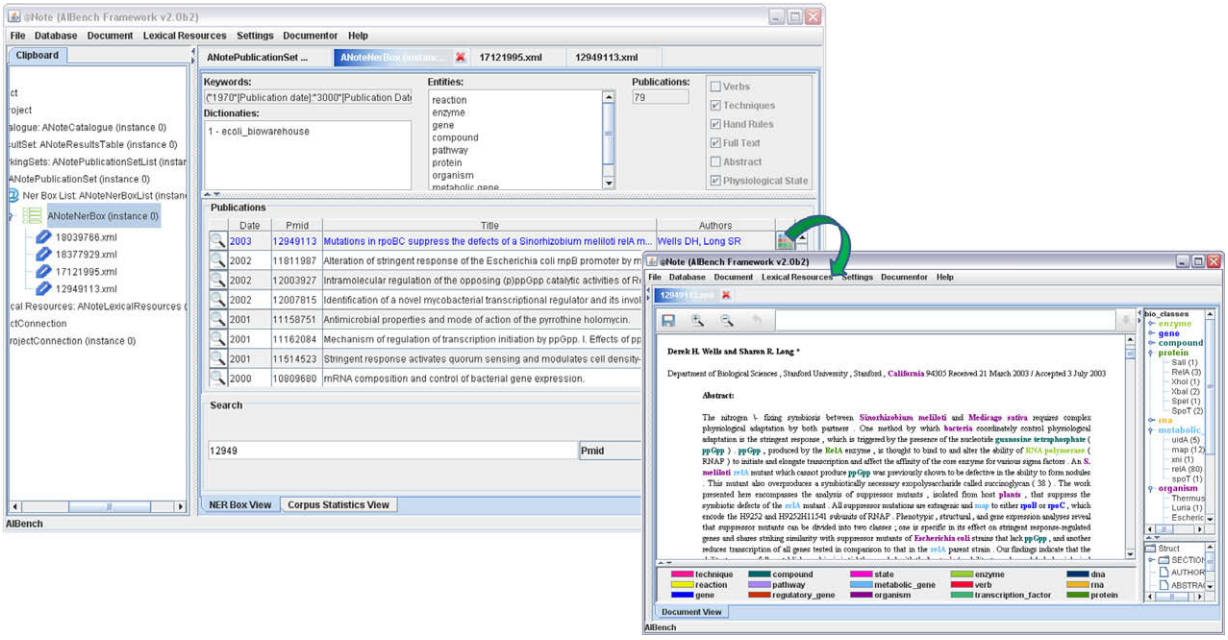


Fig. 6. Visualisation of document annotations within @Note.

amount of documents already in catalogue, available Internet bandwidth (retrieval speed) and the performance of journal servers (web site and catalogue administration). Nevertheless, the process is completely transparent to the user that will only acknowledge the set of documents outputted for the query.

PDF conversion into plain text and basic document structuring are also transparent to the user that will only see the documents already in HTML format. Biologist intervention is requested only at L-NER, where he configures the process by selecting the most adequate dictionary, and eventually, some additional lookup tables and rules. The user can choose one of the existing dictionaries (like the *E. coli* dictionary we have built from a Biowarehouse repository) or may deploy the construction of a particular one from the current set of supported data sources. Also, if using a multi-class dictionary, the user may specify the subset of supported biological classes to be annotated (for example, only genes and proteins).

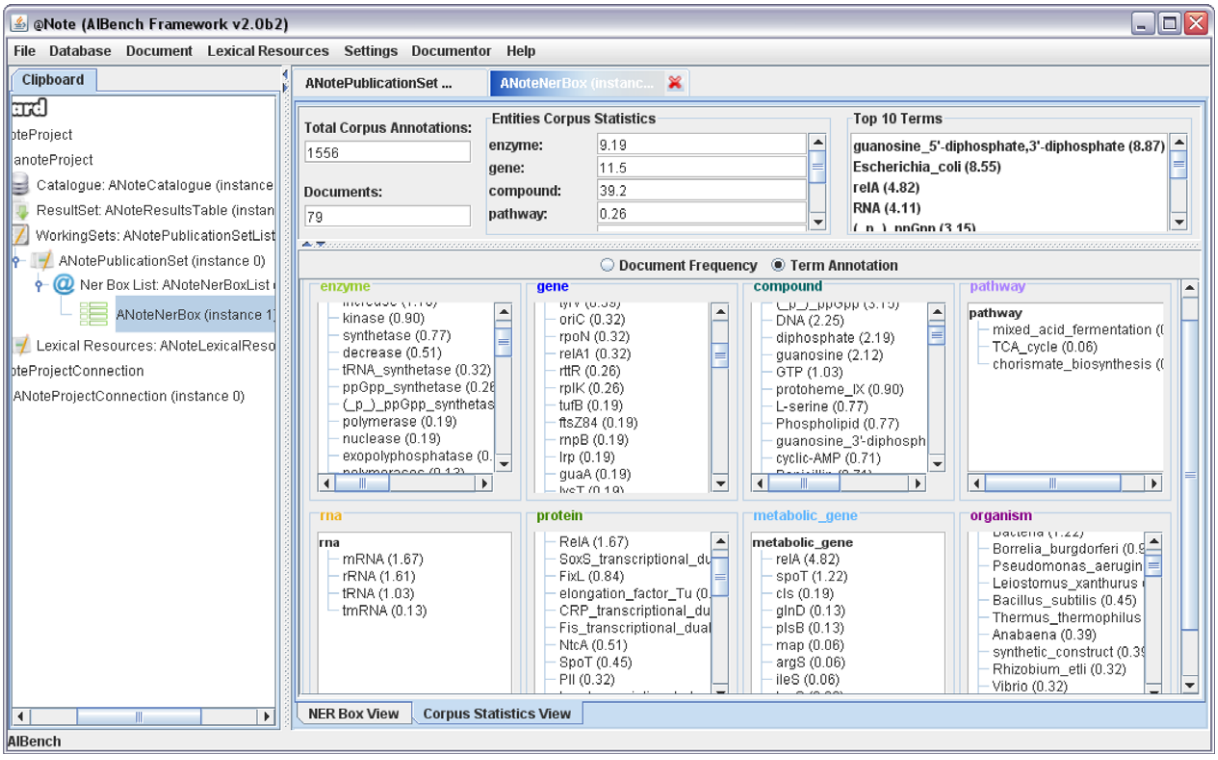


Fig. 7. Statistics visualisation. The application supports the visualisation of main corpus statistics. Namely, it displays the top 10 most annotated terms, term frequencies by biological class and document frequencies by biological class.

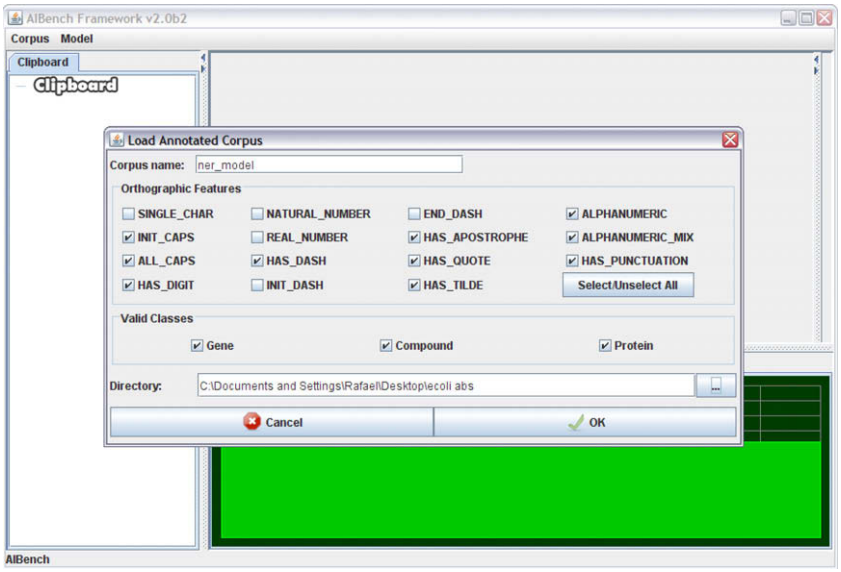


Fig. 8. Loading the corpus and preparing the NER training set. The revised corpus is used by the text miner to prepare a ML-based NER model. Dataset preparation involves the orthographic characterization of the annotated terms as well as similar processes over term's affixes and contexts.

Automatically annotated documents (Fig. 6) are made available to the user in the manual curation environment for expert revision. In this environment, identified terms appear in different colours, depending on their class, facilitating the view of the annotation performed. When correcting or adding a new annotation, the biologist uses expert knowledge to deal with dictionary incompleteness and eventual inconsistencies. Moreover, problems such as 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.

The biologist is also able to search for additional information about text mentions (already annotated or not) in Web-accessible databases such as UniProt or Biocyc, accounting for unfamiliar terminology and confirming some classifications.

After performing expert revision or immediately after automated L-NER, the user has the opportunity to look at some statistics (Fig. 7) regarding, for example, the frequencies of a given entity within the documents analysed.

3.2. The text miner perspective – preparing a particular NER model

All the operations mentioned in this section are provided in the @Note Mining application available in the project's web site. Again, the detailed steps of this task are described in Supplementary Material. After the annotated corpus is revised by the biologist, the text miner can perform mining experiments, evaluating which features and algorithms are more suitable for the NER problem.

First, the text miner prepares the training dataset (Fig. 8), processing the orthographic features (such as the presence of uppercase letters, digits, dashes, etc.) of each annotated term, as well as the corresponding affix elements (ranging from 3 to 5 character-long) and context information (2 words for each side). Also, he decides which class(es) the model will learn to classify.

The statistical characterization of the annotated corpus is an important support for the work of the text miner (Fig. 7). Document frequency and term annotation reports indicate the representativeness of the corpus regarding each biological class and its overall balance.

The deployment of text mining experiments implies the parameterisation of ML algorithms and further evaluation of the obtained model. At this level, @Note supports full algorithm parameterisation and *k*-fold cross-validation.

After several experiments, the text miner delivers the ML-based NER model that he considers most adequate for the given problem and the biologist has then the possibility of substituting the L-NER module by the M-NER module in the curation process. Thus, biologist and miner cooperate in the overall task of gathering information about a particular problem ("*E. coli* stringent response"), accounting for expert domain knowledge and mining skills, respectively.

4. Conclusions

The @Note project aims at fulfilling the existing gap between BioTM researchers 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 in managing and processing the ever growing literature. Furthermore, it accounts for BioTM research needs, providing means for experts to prepare and deploy NLP and ML experiments using well-known tools such as GATE, YALE and WEKA. Also, 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 and 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, an invaluable asset for BioTM research.

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

Availability

The project is made available, together with documentation and other resources, in the project home page given below.

More details:

- Project name: @Note Biomedical Text Mining workbench.
- Project home page: http://sysbio.di.uminho.pt/anote/wiki.
- Operating system(s): Platform independent.

- Programming languages: Java and Perl.
- Other requirements: Java JRE 1.5 or higher, Strawberry Perl 5.10 (required for Windows only!), Xpdf 3.02 (required for Windows or Linux), pdftotext (required for Mac OS) and MySQL server 5.1.
- License: GNU-GPL, version 3.

Acknowledgments

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

The work of SAC is supported by a PhD grant from the Fundação para a Ciência e Tecnologia (ref. SFRH/BD/22863/2005).

References

[1] Kersey P, Apweiler R. Linking publication, gene and protein data. Nat Cell Biol 2006;8:1183–210.
[2] Zweigenbaum P, mner-Fushman D, Yu H, Cohen KB. Frontiers of biomedical text mining: current progress. Brief Bioinform 2007;8:358–75.
[3] Ananiadou S, Kell DB, Tsujii JI. Text mining and its potential applications in systems biology. Trends Biotechnol 2006;24:571–9.
[4] Natarajan J, Berrar D, Hack CJ, Dublitzky W. Knowledge discovery in biology and biotechnology texts: a review of techniques, evaluation strategies, and applications. Crit Rev Biotechnol 2005;25:31–52.
[5] Erhardt RAA, Schneider R, Blaschke C. Status of text-mining techniques applied to biomedical text. Drug Discov Today 2006;11:315–25.
[6] Tsai RT, Sung CL, Dai HJ, Hung HC, Sung TY, Hsu WL. NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. BMC Bioinform 2006;7(Suppl. 5):S11.
[7] Schmeier S. Automated recognition and extraction of entities related to enzyme kinetics from text. Freie Universität Berlin; 2005.
[8] Crim J, McDonald R, Pereira F. Automatically annotating documents with normalized gene lists. BMC Bioinform 2005;6(Suppl. 1):S13.
[9] Sun CJ, Guan Y, Wang XL, Lin L. Biomedical named entities recognition using conditional random fields model. Fuzzy Syst Knowledge Discov Proc 2006;4223:1279–88.
[10] Abi-Haidar A, Kaur J, Maguitman A, Radivojac P, Retchsteiner A, Verspoor K, et al. Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks. Genome Biol 2008.
[11] Chang JT, Altman RB. Extracting and characterizing gene–drug relationships from the literature. Pharmacogenetics 2004;14:577–86.
[12] Palakal M, Stephens M, Mukhopadhyay S, Raje R, Rhodes S. A multi-level text mining method to extract biological relationships. Proc IEEE Comput Soc Bioinform Conf 2002;1:97–108.
[13] Chen D, Muller HM, Sternberg PW. Automatic document classification of biological literature. BMC Bioinform 2006;7.
[14] Hao PY, Chiang JH, Tu YK. Hierarchically SVM classification based on support vector clustering method and its application to document categorization. Expert Syst Appl 2007;33:627–35.
[15] Afantenos S, Karkaletsis V, Stamatopoulos P. Summarization from medical documents: a survey. Artif Intell Med 2005;33:157–77.
[16] Chiang JH, Liu HS, Chao SY, Chen CY. Discovering gene–gene relations from sequential sentence patterns in biomedical literature. Expert Syst Appl 2007;33:1036–41.
[17] Hoffmann R, Valencia A. Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics 2005;21(Suppl. 2):ii252–8.
[18] Chen H, Sharp BM. Content-rich biological network constructed by mining PubMed abstracts. BMC Bioinform 2004;5.
[19] Liu Y, Navathe SB, Civera J, Dasigi V, Ram A, Ciliax BJ, et al. Text mining biomedical literature for discovering gene-to-gene relationships: a comparative study of algorithms. IEEE ACM Trans Comput Biol Bioinform 2005;2:62–76.
[20] Karopka T, Scheel T, Bansemer S, Glass A. Automatic construction of gene relation networks using text mining and gene expression data. Med Inform Internet Med 2004;29:169–83.
[21] Chaussabel D, Sher A. Mining microarray expression data by literature profiling. Genome Biol 2002;3.
[22] Hokamp K, Wolfe KH. PubCrawler: keeping up comfortably with PubMed and GenBank. Nucleic Acids Res 2004;32:W16–9.
[23] Rebholz-Schuhmann D, Kirsch H, Arregui M, Gaudan S, Riethoven M, Stoehr P. EBIMed – text crunching to gather facts for proteins from Medline. Bioinformatics 2007;23:E237–44.
[24] PubMed [http://www.ncbi.nlm.nih.gov/pubmed].
[25] Cohen KB, Hunter L. Natural language processing and systems biology. In: Dubitzky, Pereira, editors. Artificial intelligence methods and tools for systems biology. Springer Verlag; 2004.
[26] Karamanis N, Seal R, Lewin I, McQuilton P, Vlachos A, Gasperin C, et al. Natural language processing in aid of FlyBase curators. BMC Bioinform 2008;9.
[27] Tsuruoka Y, Tateishi Y, Kim JD, Ohta T, McNaught J, Ananiadou S, et al. Developing a robust part-of-speech tagger for biomedical text. Adv Inform Proc 2005;3746:382–92.
[28] Olsson F, Eriksson G, Franzén K, Asker L, Lidén P. Notions of correctness when evaluating protein name taggers. In: Proceedings of COLING 2002, Taipei, Taiwan; 2002.
[29] Smith L, Rindflesch T, Wilbur WJ. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics 2004;20:2320–1.
[30] 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 J Res Dev 2004;48:693–701.
[31] Chen L, Liu H, Friedman C. Gene name ambiguity of eukaryotic nomenclatures. Bioinformatics 2005;21:248–56.
[32] Fundel K, Zimmer R. Gene and protein nomenclature in public databases. BMC Bioinform 2006;7.
[33] Spasic I, Ananiadou S, McNaught J, Kumar A. Text mining and ontologies in biomedicine: making sense of raw text. Brief Bioinform 2005;6:239–51.
[34] Liu HF, Hu ZZ, Torii M, Wu C, Friedman C. Quantitative assessment of dictionary-based protein named entity tagging. J Am Med Inform Assoc 2006;13:497–507.
[35] Hu ZZ, Narayanaswamy M, Ravikumar KE, Vijay-Shanker K, Wu CH. Literature mining and database annotation of protein phosphorylation using a rule-based system. Bioinformatics 2005;21:2759–65.
[36] Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J. ProMiner: rule-based protein and gene entity recognition. BMC Bioinform 2005;6.
[37] Regev Y, Finkelstein-Landau M, Feldman R. Rule-based extraction of experimental evidence in the biomedical domain: the KDD Cup 2002 (task 1). SIGKDD Explor Newsl 2002;4:90–2.
[38] Yeganova L, Smith L, Wilbur WJ. Identification of related gene/protein names based on an HMM of name variations. Computat Biol Chem 2004;28:97–107.
[39] Tsuruoka Y, Tsujii J. Improving the performance of dictionary-based approaches in protein name recognition. J Biomed Inform 2004;37:461–70.
[40] Dimililer N, Varoglu E. Recognizing biomedical named entities using SVMs: improving recognition performance with a minimal set of features. Knowledge Discov Life Sci Lit Proc 2006;3886:53–67.
[41] Pahikkala T, Ginter F, Boberg J, Jarvinen J, Salakoski T. Contextual weighting for support vector machines in literature mining: an application to gene versus protein name disambiguation. BMC Bioinform 2005;6.
[42] Kim JD, Ohta T, Tateisi Y, Tsujii J. GENIA corpus – semantically annotated corpus for bio-textmining. Bioinformatics 2003;19(Suppl. 1):i180–2.
[43] Hirschman L, Yeh A, Blaschke C, Valencia A. Overview of BioCreAtIvE: critical assessment of information extraction for biology. BMC Bioinform 2005;6(Suppl. 1):S1.
[44] Hersh W, Bhupatiraju RT, Ross L, Johnson P, Cohen AM, Kraemer DF. TREC 2004 Genomics Track Overview. 13–31.
[45] Hersh W, Bhupatiraju RT. TREC Genomics Track Overview. 14–23.
[46] Callisto [http://callisto.mitre.org/].
[47] Morton T, LaCivita J. WordFreak: an open tool for linguistic annotation. NJ, USA. 17–18.
[48] MMax2 [http://mmax.eml-research.de/].
[49] Cohen AM, Hersh WR. A survey of current work in biomedical text mining. Brief Bioinform 2005;6:57–71.
[50] Cunningham H. GATE, a general architecture for text engineering. Comput Humanit 2002;36:223–54.
[51] Rapid-I [http://rapid-i.com/].
[52] Entrez programming utilities [http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html].
[53] Medical subject headings [http://www.nlm.nih.gov/mesh/].
[54] LWP::Simple – simple procedural interface to LWP [http://search.cpan.org/~gaas/libwww-perl-5.810/lib/LWP/Simple.pm].
[55] WWW-Mechanize [http://search.cpan.org/dist/WWW-Mechanize/].
[56] Xpdf [http://www.foolabs.com/xpdf/].
[57] pdftotext [http://www.bluem.net/downloads/pdftotext_en/].
[58] PDFBox [http://www.pdfbox.org/].
[59] Lourenço A, Carneiro S, Carreira R, Rocha M, Rocha I, Ferreira EC. A tool for the automatic and manual annotation of biomedical documents. 85–92.
[60] Text-RewriteRules-0.11 [http://search.cpan.org/~ambs/Text-RewriteRules-0.11/].
[61] BioCyc database collection [http://www.biocyc.org/].
[62] UniProt – the universal protein resource [http://www.uniprot.org/].
[63] Chemical entities of biological interest (ChEBI) [http://www.ebi.ac.uk/chebi].
[64] The NCBI taxonomy [http://www.ncbi.nlm.nih.gov/Taxonomy/].
[65] BioWarehouse – database integration for bioinformatics [http://biowarehouse.ai.sri.com/].
[66] Frank E, Hall M, Trigg L, Holmes G, Witten IH. Data mining in bioinformatics using Weka. Bioinformatics 2004;20:2479–81.
[67] Witten IH, Frank E. Data mining: practical machine learning tools and techniques. 2nd ed. San Francisco: Morgan Kaufman; 2005.
[68] AI Bench [http://www.aibench.org].
[69] Mukherjee TK, Raghavan A, Chatterji D. Shortage of nutrients in bacteria: the stringent response. Curr Sci 1998;75:684–9.