BioEve: User Interface Framework Bridging IE and IR

by

Pradeep Kanwar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2010 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Valentin Dinu
Baoxin Li

ARIZONA STATE UNIVERSITY

December 2010

ABSTRACT

Continuous advancements in biomedical research have resulted in the production of vast amounts of scientific data and literature discussing them. The ultimate goal of computational biology is to translate these large amounts of data into actual knowledge of the complex biological processes and accurate life science models. The ability to rapidly and effectively survey the literature is necessary for the creation of large scale models of the relationships among biomedical entities as well as hypothesis generation to guide biomedical research. To reduce the effort and time spent in performing these activities, an intelligent search system is required. Even though many systems aid in navigating through this wide collection of documents, the vastness and depth of this information overload can be overwhelming. An automated extraction system coupled with a cognitive search and navigation service over these document collections would not only save time and effort, but also facilitate discovery of the unknown information implicitly conveyed in the texts. This thesis presents the different approaches used for large scale biomedical named entity recognition, and the challenges faced in each. It also proposes BioEve: an integrative framework to fuse a faceted search with information extraction to provide a search service that addresses the user's desire for "completeness" of the query results, not just the top-ranked ones. This information extraction system enables discovery of important semantic relationships between entities such as genes, diseases, drugs, and cell lines and events from biomedical text on MEDLINE, which is the largest publicly available database of the world’s biomedical journal literature. It is an innovative search and discovery service that makes it easier to search/navigate and discover knowledge hidden in life sciences literature. To demonstrate the utility of this system, this thesis also details a prototype enterprise quality search and discovery service that helps researchers

with a guided step-by-step query refinement, by suggesting concepts enriched in intermediate results, and thereby facilitating the "discover more as you search" paradigm.

ACKNOWLEDGMENTS

I extend my sincere gratitude and appreciation to all the people who made this master's thesis possible. I would like to take this opportunity to thank my advisor, Dr. Hasan Davulcu, for guidance and encouragement during my masters and while writing this thesis. My sincere thanks to Dr. Valentin Dinu and Dr. Baoxin Li for being on my thesis committee and for providing the guidance and the feedback on my work. I would also like to specially thank Syed Toufeeq Ahmed for his ideas, encouragement and support that helped me make this thesis more structured and comprehensive. Many thanks to all my lab-mates for making this journey so exciting. I am very grateful for the love and the unconditional support of my family and friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The discovery of the human gene and rapid developments in the biomedical domain has produced large amounts of genetic data resulting in exponential growth of biomedical literature. In recent years efforts to analyze this data has been correspondingly constrained by the challenge of organizing and analyzing it. The urgency of this task and the reward of even partial success in its accomplishment have caused the development and interoperability between diverse web-based representations to take center stage. Much of the valuable knowledge gained is found in published articles, journals and thus in un-structured textual form. MEDLINE, the primary research database currently contains more than 20 million[1] abstracts covering from 1948-present. The growth rate of MEDLINE database is expected to be 400,000 articles per year (Ulf Leser 2005). This data makes the task of expert database curators and reviewers, to recognize and discover important connections between biomedical entities, very tedious and time consuming. There is a need to understand user interest and retrieve novel information, which may otherwise be obscured by the sheer volume of biomedical literature. Tools to help researchers achieve this while coping with the information overload are therefore the solution. The BioEve Discovery Engine[2] is an innovative search and discovery service that makes it easier to search/navigate and explore (Ryen W. White 2006) knowledge hidden in life sciences literature.

Bio-Eve's intuitive and cognitive interface enables discovery of important semantic relationships between entities like drugs, diseases, and genes; it highlights these to truly

[1] http://www.cas.org/ASSETS/BF043DBCE4274170A03561C274C671D2/medline.pdf

[2] BioEve discovery engine received 2010 ASU Innovation Challenge grant for innovation and impact,See http://innovationchallenge.asu.edu/winners2010.html and http://asunews.asu.edu/20100506_innovationchallenge

facilitate the "discover more as you search" paradigm. BioEve can be accessed here[3]:(http://www.bioeve.org/, Use Firefox/Chrome/Safari browser for faster experience), and currently has 1,908,682 abstracts annotated, classified, and indexed. Human genome sequencing marked the beginning of the era of large-scale genomics and proteomics, leading to large quantities of information on sequences, interactions, and their annotations. Many experimental findings are reported in the -omics literature, where researchers have access to over 20 million publications, with 2,000 to 4,500 new ones every day (source: PubMed citation index[4]).

This vast increase in available information demands novel strategies to help researchers to keep up-to-date with recent developments. Tools should provide dedicated and intuitive strategies that help to find relevant literature, starting with initial keyword searches and drilling down results via overviews enriched with auto generated suggestions to refine queries. BioEve (http://www.bioeve.org/) provides semantic faceted search with real time query refinement, where users can quickly refine their queries and drill down to the articles they are looking for in a matter of seconds, corresponding to a few number of clicks. BioEve can identify hidden important semantic relationships between entities like drugs, diseases, and genes, and highlight them to enable "discover more as you search."

[3] Firefox/Chrome/Safari/Opera browsers are recommended for a faster experience

[4] PubMed: http://www.pubmed.gov

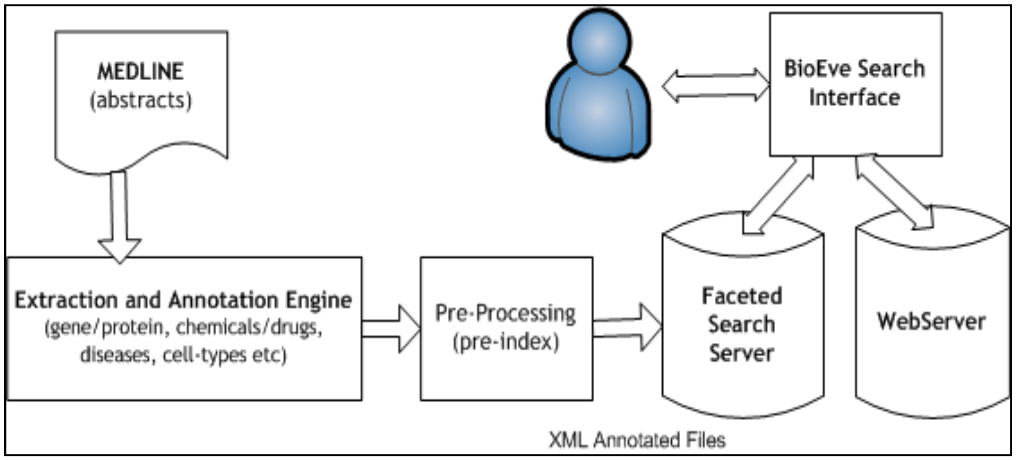

Figure 1.1 BioEve system overview

1.1 MOTIVATION

The motivation behind this thesis work is to develop a system which can identify hidden relationships between entities like drugs, diseases and genes and highlights them, thereby allowing the researcher to not only navigate the literature, but also to see entities and the relations they are involved in immediately, without having to read the article text fully, thus providing another aspect of searching relevant articles. Figure 1.2 illustrates the motivation for this work which is to facilitate the following paradigms:

Knowing what keywords to search

How to refine and narrow large set of results returned

Insights into important relationship will be of tremendous help

Faster and easier way to locate relevant articles with fewer clicks than reading pages of text

Discover more as you search, and refine or expand search on the fly with dynamic interface

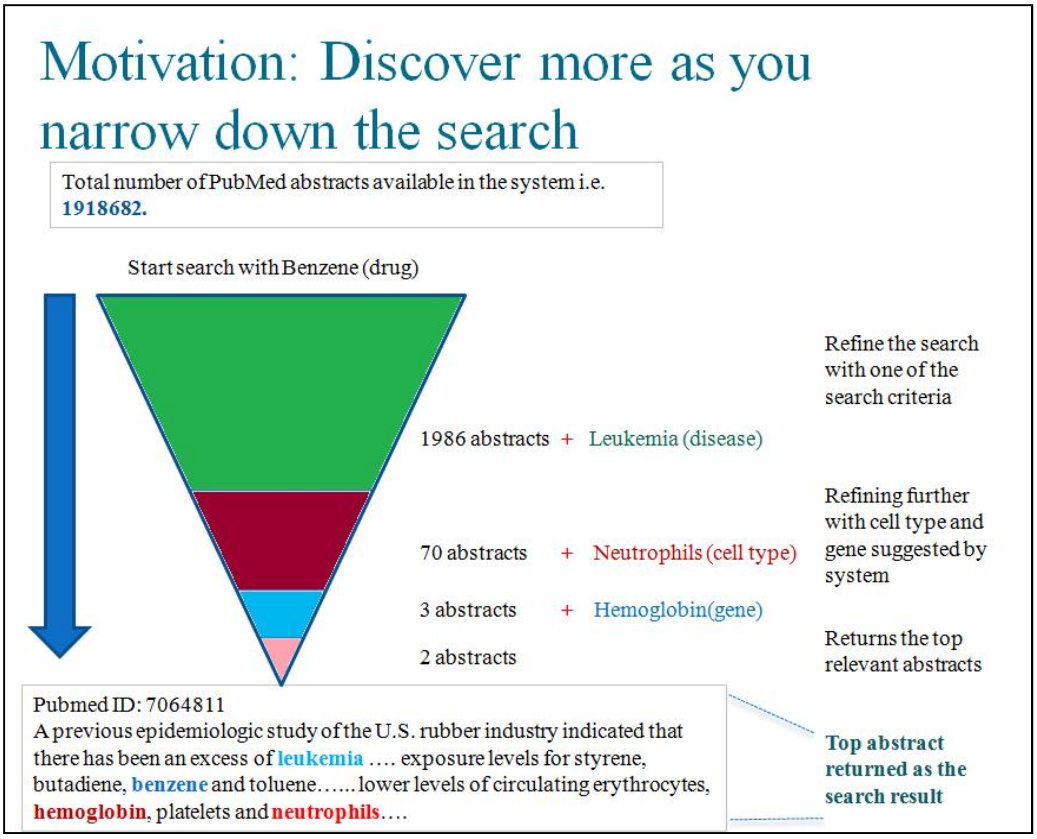

Figure 1.2 Motivation

The figure shows the total number of abstracts (1918682) taken from PubMed[5] database maintained by United States National Library of Medicine. The PubMed database has a rich and up to date collection of biomedical literature, and is used by a lot of applications as well as researchers in biomedical domain. Tools should provide dedicated and intuitive strategies that help to find relevant literature, starting with initial keyword searches and drilling down results via overviews enriched with auto-generated suggestions to refine queries. BioEve (http://www.bioeve. org/) provides semantic faceted search with real time query refinement, where users can quickly refine their queries and drill down to the articles they are looking for in a matter of seconds, to whom correspondence should be addressed corresponding to a few number of clicks. BioEve can identify hidden important

[5] http://www.ncbi.nlm.nih.gov/pubmed

semantic relationships between entities like drugs, diseases, and genes, and highlight them to enable discover more as you search. Consider an example where the researcher wants to study the relation between *benzene, leukemia* and N*eutrophils cells*. Need is to find all the top relevance abstracts about the experimental studies explaining the relation between these two entities. As shown in the figure, the number of abstracts which talks about *benzene* is 1986. Searching further, the users would want to go through top relevant abstracts for the given query. As shown, the disease *Leukemia* selection narrows down the resulting abstract set to 70. Going further ahead in making more filter choices, by including *Neutrophils* cell types in the filter criteria narrows it further down to 3 abstracts. Required is the system which suggests top relevant and frequently occurring bio medical entities meaningful to the user's search query which would be full filled further when the system suggests *Hemoglobin* as the top relevant gene reducing the number of top relevant abstracts count to 2.Thus, the aim here is to reduce the number of abstracts for researchers to top relevant ones to through while studying about entity relationships. Additionally, the aim is also to implement the paradigm "discover more as you search" by identifying other important relationship among entities which otherwise goes unnoticed.

1.2 THESIS OUTLINE

The primary objective of this work is to develop a framework providing web based search interface in biomedical field, which would help researchers and scientists in faster discovery of connections and patterns between different biomedical entities. After identifying the entity types like drugs, diseases, chemicals, gene and proteins, we index them on Solr to aid in performing faceted search. The major contribution of this work is to identify biomedical entities and phrases indicating their connections and facilitate users to discover and study these connections as well as discover new patterns among

these entities. Chapter 2 discusses the related work supporting our approach. Chapter 3 describes the BioEve: novel integrative system architecture in detail, explaining each building block of this framework. We recognized different types of bio entities and biomedical terms and tagged them to facilitate faceted search. Different entities tagging approaches are elaborately discussed in Chapter 4.

. Chapter 5 describes the annotation mechanism for bio entities and events. Chapter 6 details the BioEve: web-based user interface and guidelines on how to use it and the user feedbacks and comments about this system. Chapter 7 summarizes this work and describes the future work to be done on this BioEve system.

Chapter 2

BACKGROUND AND RELATED WORK

The vast amount of biomedical information available on shared systems like PubMed requires a way to identity hidden relationships between entities like drugs, diseases and genes and highlights them, thereby allowing the researcher to not only navigate the literature, but also to see entities and the relations they are involved in immediately, without having to read the article text fully, thus providing another aspect of searching relevant articles. The existing web-based search interfaces provide the bridge between Information Extraction (IE) and Information Retrieval (IR). However, current need is to assist the researchers and scientists in faster, smarter searches and pattern discoveries while studying the existing knowledge base. Information extraction (IE) is a process which selectively structures and combines data found implicitly or explicitly in one or more texts (for e.g. biomedical literature) including extracting entities, relations, and events. Typical key areas of information extraction in biomedical domain research are named entity recognition, co-reference resolution, terminology extraction, and relationship extraction (for e.g. extracting protein - protein interactions)[6].

2.1 INFORMATION EXTRACTION FROM BIOMEDICAL TEXT

The last two decades have seen unprecedented growth in both the production of biomedical data and amount of published literature discussing it. Advances in computational and biological methods have remarkably changed the scale of biomedical research. Complete genomes can now be sequenced within months and even weeks (Shatkay 2005), computational methods expedite the identification of tens of thousands of genes and large-scale experimental methods. The main developments in this area have been related to the identification of biological entities (named entity recognition), such as

[6] http://en.wikipedia.org/wiki/Information_extraction

protein and gene names in free text, the association of gene clusters obtained by microarray experiments with the biological context provided by the corresponding literature, automatic extraction of protein interactions and associations of proteins to functional concepts (e.g. gene ontology terms). Using conventional methods for finding implicit and explicit information from this huge collection of literature is very difficult. Hence the need for new information extraction methods in biomedical text mining is needed. Though scientists in the field are aided by many online databases of biochemical interactions, currently a majority of these are labor intensively curated by domain experts. Information extraction from text has therefore been pursued actively as an attempt to extract knowledge from published material and to speed up the curation process significantly. The most used online source of biomedical literatures is PubMed/MEDLINE[7] database, which is maintained by National Center of Biotechnology Information (NCBI) and contains over 20 million scientific abstracts.

Two of the main obstacles in the way of fully automatic extraction of facts from free-form natural language text are ambiguity and variability. For example, even a simple relationship such as one protein binding another can be expressed in a surprising number of ways; twenty two ways are shown in the table below (Pyysalo 2008):

[7] http://www.ncbi.nlm.nih.gov/pubmed/

1. e1 binds e2	12. e1 has been implicated in e2 binding
2. e1 cross-links e2	13. e1 binding region of e2
3. binding of e1 to e2	14. e1 is secreted as a protein that binds
4. e1 binding to e2	e2
5. binding to e1 by e2	15. affinity of e1 for e2
6. e1 is able to bind to e2	16. e1 regulates e3 by binding to e2
7. e1 is shown to bind e2	17. association of e1 with e2
8. e1 is an antigen known to bind e2	18. e1 is directly associated with e2
9. e1 (an e2-binding protein)	19. e1, which binds to e2,
10. partners that associate with e1: e2 and e3	20. e1 is a receptor for e2
11. e1 is involved in binding to e2	21. e1 binding sites of e2
	22. e1 is expressed as a receptor for e2

Table 2.1 Protein-Protein Binding relationships

Most efforts, concerned with information extraction in biomedical domain, are focused in using curated lexica or natural language processing for identifying relevant phrases and facts in text. Information extraction from free-text utilizes shallow-parsing techniques (Walter Daelemans 1999), Parts-of-Speech (Brill 1992), noun and verb phrase chunking (Andrei Mikheev 1997), verb subject and object relationships (Walter Daelemans 1999), and learned (Mary Elaine Califf 1999), (Mark Craven 1999), (Seymore K. 1999) or hand-build patterns to automate the creation of specialized databases. Manual pattern engineering approaches employ shallow parsing with patterns to extract the interactions. In the (Ono T 2001) system, sentences are first tagged using a dictionary based protein name identifier and then processed by a module which extracts interactions directly from complex and compound sentences using regular expressions based on part of speech tags. IE systems look for entities, relationships among those entities, or other specific facts

within text documents. The success of information extraction depends on the performance of the various subtasks involved.

In the biomedical context, the first step toward information extraction is to recognize the names of proteins (Fukuda K 1998), genes, drugs and other molecules. The next step is to recognize interaction events between such entities (Blaschke C 1999), (Christian Blaschke 2002), (Thomas J 2000), (Ono T 2001), (Udo Hahn 2002)and then to finally recognize the relationship between interaction events.

With advent of Multi-core machines, and cheaper hardware cost, parallelization and distributed processing are attractive alternatives for processing extremely large collections. Information extraction is particularly amenable to parallelization, as the main information extraction steps, (e.g., part-of-speech tagging, shallow syntactic parsing, named entity recognition) operate over each document independently (Stephen Dill 2003). Most parallel data mining and distributed processing architectures (e.g., Google's MapReduce[8] or Apache Hadoop[9] can be easily adapted for information extraction over large collections.

Large scale systems for information extraction include many different classifiers and extractors. In systems containing may learned components, it is important to cleanly share information between the components and to flexibly sequence the actions of the components, (Cohen 2003), discuss how to cleaning share information between components.

2.2 TEXT CATEGORIZATION

Text Categorization, is labeling of natural language texts with thematic categories from a predefined set of category-tags. There are two main approaches: Knowledge Engineering

[8] MapReduce: http://labs.google.com/papers/mapreduce.html

[9] Hadoop: http://hadoop.apache.org/

and Machine Learning. While Knowledge Engineering approaches rely on a domain expert to specify the classification rules, machine learning approaches are automated and prevalently use various clustering and classification algorithms. Conventional classification algorithms are often augmented using feature selection procedures that enhance the categorization of documents. Word co-occurrences with ontologies used for the semantics of the words are also used in the classification of documents. The CONSTRUE (Philip J. Hayes 1990) system follows the knowledge engineering approach, where the rules are specified as a disjunction of conjunctive clauses. The machine learning (ML) approach need existence of a training set of documents, already classified into set of categories. There are two kinds of ML-based categorization, known as hard and soft classification. Hard classification assigns a truth value (either True or False) to each document, where as soft classification gives a ranking (by relevance) to each document.

2.3 NAMED ENTITY RECONGITION

Entity extraction or Named entity identification is the process of identifying the words or phrases of interest such as genes, proteins, protein families, drugs, chemicals and pathways in text. Entity identification has also been thoroughly researched over the years, with various challenges such as the BioCreative[10] and shared tasks in conferences addressing the issues and evaluating the performances using common corpora. The simplest and frequently used approach to entity identification is a dictionary matching approach where the entity names are compiled as a dictionary and a string match with an entry in the dictionary tags the words or phrases as gene or protein names.

[10] http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html

A variety of publicly available databases provide the resources for entity names. NCBI's LocusLink[11] and HUGO[12] are among the databases that provide gene and protein names and their synonyms.

Entity Identification systems generally use rule based approaches and machine learning techniques to mark the phrases of interest in text. Rule based approaches rely on regular expressions and heuristic rules to identify gene names. Fukuda et al. follow a combination of regular expressions and expansion rules to identify single word and multi-word gene names. Some of the machines learning approaches followed for NER include decision trees, Bayesian classifiers, hidden markov models, iterative error reduction, boosted wrapper induction and support vector machines. The ABGene system from Tanabe and Wilbur (Lorraine Tanabe 2002) uses the Brills tagger to learn transformation rules to tag the gene and protein names in text. The rules are based on the word occurrences, neighboring words and part of speech tags of the words and the neighbors. Research in entity recognition has resulted in the development of various corpora for the purpose of providing a benchmark for the entity recognition systems. The GENIA corpus, a hand -annotated corpus of abstracts from over 2000 MEDLINE articles on human blood transcription factors uses the GENIA ontology to tag concepts in text. ABNER (Settles 2005)system uses Conditional Random fields (CRF), complemented by a set of orthographic and semantic features to identify proteins, genes, cell lines and cell types. Semantic features were in the form of hand curated lexicons and ones drawn from databases and Google data sets. Addition of these semantic features worked well for classification of low frequency classes in the corpus.

[11] http://www.ncbi.nlm.nih.gov/pro jects/LocusLink/

[12] http://www.gene.ucl.ac.uk/nomenclature/

2.4 INFORMATION RETRIEVAL IN BIOMEDICAL INFORMATICS

With the advancement of genomic and proteomic technologies, the increased number of publications discussing genes and proteins leads to the ultimate goal of modern large scale biology to translate these large amounts of data into actual knowledge of biological processes for text mining and interpretation and planning of large scale experiments. Information retrieval (IR) (Shatkay 2005) is a necessary first step towards text mining. It is the process of identifying a subset of documents accurately and efficiently whose content is most relevant to a user's need within a large collection. Various approaches of IR includes Boolean queries, index structures, similarity based approaches and text categorization. PubMed applies Boolean query to its search feature.

In the large-scale genomics, many methods to support IR have been introduced and developed for e.g. Swanson's 'transitive' relations theory i.e. indirect links among entities as clues for yet unknown relationships. This reasoning was further studied and automated by Weeber and Srinivasan and Libbus and Wren. Also, Shatkay introduced thematic analysis methods for finding functional relationships among genes (Shatkay 2005).

However, since information retrieval does not look for explicitly stated information in literature, it has the disadvantage of foreshadowing the undiscovered facts. Therefore, to successfully mine the biomedical literature, it is important to relate the merits and limitations of the different IR methods.

2.5 KEYWORD AND FACETED SEARCH

PubMed is one of the most well known and used citation indexes for the Life Sciences. It provides basic keyword searches and benefits largely from a hierarchically organized set of indexing terms, MeSH, that are semi-automatically assigned to each article. Screenshot of PubMed Interface is shown in figure 2.1. PubMed also enables quick searches for

related publications (Bénel A. 2002) given one or more articles deemed relevant by the user. A few commercial products are currently available that provide additional services, but they also rely on basic keyword search, with no real discovery or dynamic faceted search. Examples are OvidSP[13] and Ingenuity Answers[14], both of which support book-marking as one means of keeping track of visited citations.

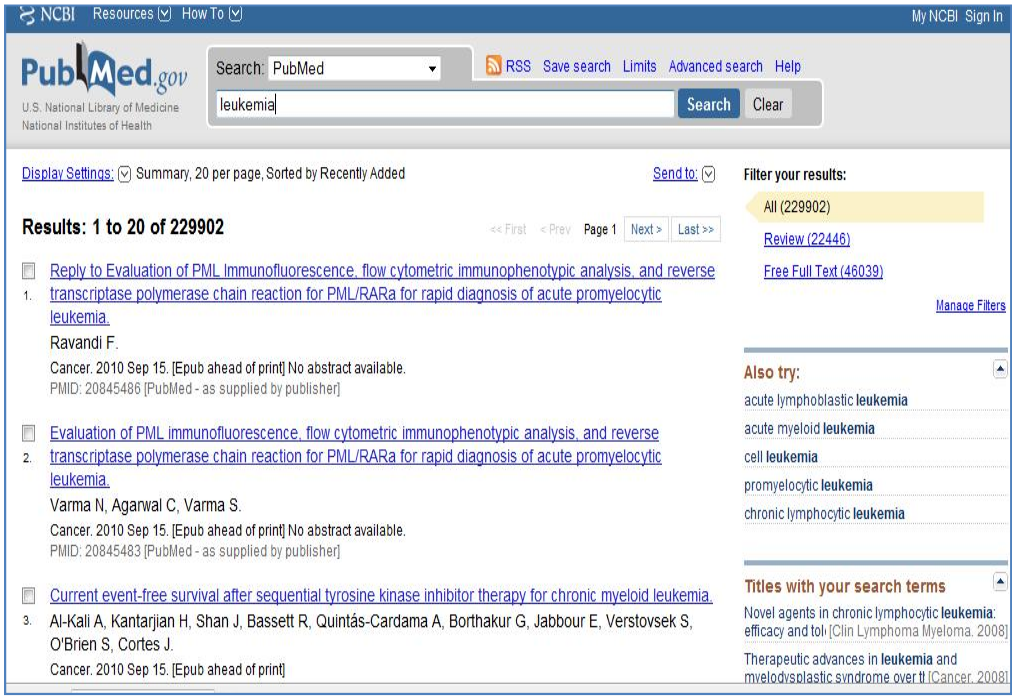

Figure 2.1: Screenshot of PubMed; Results for query “leukemia”.

Research tools such as EBIMed[15] provide additional cross-referencing of entities to databases such as UniProt[16] or to the GeneOntology[17]. They also try to identify relations between entities of the same or different types, such as protein-protein interactions,

[13] http://www.ovid.com/site/resources/index_ovidsp.jsp

[14] http://www.ingenuity.com/products/answers.html

[15] EBIMed: http://www.ebi.ac.uk/Rebholz-srv/ebimed/index.jsp

[16] http://www.uniprot.org/

[17] http://www.geneontology.org/

functional protein annotations, or gene–disease associations. GoPubMed[18] guides users in their everyday searches by mapping articles to concept hierarchies, such as the Gene Ontology and MeSH (Andreas Doms 2005). For each concept found in abstracts returned EBIMed by the initial user query, GoPubMed computes a rank based on occurrences of that concept. Thus, users can quickly grasp which terms occur frequently, providing clues for relevant topics and relations, and refine subsequent queries by focusing on particular concepts, discarding others.

BioEve can identity hidden relationships between entities like drugs, diseases and genes and highlights them, thereby allowing the researcher to not only navigate the literature, but also to see entities and the relations they are involved in immediately, without having to read the article text fully, thus providing another aspect of searching relevant articles. Screenshot of BioEve Interface is shown in figure 2.2.

Before we identity semantic relationships between entities, first important task in biomedical text mining is to recognize named entities occurring in unstructured article text, such as genes and diseases. Named entity recognition (NER) is helpful to identify relevant documents, index a document collection, and facilitate information retrieval and semantic searches (Settles 2004).

A faceted search system (or parametric search system) presents users with key value meta-data that is used for query refinement (Jonathan Koren 2008). By using facets (which are meta-data or class labels for entities such as genes or diseases), users can easily combine the hierarchies in various ways to refine and drill down the results for a given query; they do not have to learn custom query syntax or to restart their search from scratch after each refinement. Studies have shown that users prefer faceted search interfaces because of their intuitiveness and ease of use (Vineet Sinha 2005). (Hearst

[18] http://www.gopubmed.org/web/gopubmed/

2006) Hearst share their experience, best practices, and design guidelines for faceted search interfaces, focusing on supporting flexible navigation, seamless integration with directed search, fluid alternation between refining and expanding, avoidance of empty results sets, and most importantly making users at ease by retaining a feeling of control and understanding of the entire search and navigation process (Hearst 1999).

Figure 2.2 Screenshot of BioEve; Results for query “leukemia”

CHAPTER 3

BIOEVE: SYSTEM ARCHITECTURE

BioEve system (Syed Toufeeq Ahmed 2010) currently consists of 1.9 million PubMed abstracts annotated and indexed. The layered system architecture as shown in figure 3.1 can be split into four main logical layers as below:

3.1 DATA INDEXING

The data store layer has the mechanism for data indexing. MEDLINE dataset is available as zipped XML files that needed XML2text conversion and then we ingested them into an indexer for faster access and keyword based text search that allows us to select a particular subset of the abstracts for further processing. We used Apache Lucene[19] for indexing. Pubmed datasets were available to us in XML format. Our main interest was the content of the abstract and the Pubmed ID which uniquely identifies it. Since we aimed to filter a sizable number (approximately 2 million) of bio - entity rich abstracts, following were some of the challenges we faced:

• Filtering required content from a large number of abstracts was required. Faster processing technique was required to complete this in a feasible time frame.

• Storing this information as plain text would require a lot of storage space. Hence there was a need for compressed storage of this data.

• With respect to any processing in future, there may be a need for analyzing more information about an abstract. Maintaining this data should be as less cumbersome as possible.

• There should be optimized indexing on the data, so that simple keyword searches should be done in a faster manner, over this entire dataset.

[19] http://lucene.apache.org/

• Query formulation should be simple and easy to use, without having to delve into complex query languages like SQL.

To address the above challenges, we used Digester (Apache Digester n.d.), a subproject of the Jakarta Commons project, to obtain the Pubmed ID, abstract title and text. It offers a simple, high-level interface for mapping XML documents to Java objects.

3.2 BIOMEDICAL ENTITY TAGGING AND ANNOTATION

Second layer in the architecture is of Information Extraction Layer where the relevant information is extracted from the abstract text. For recognizing different gene/protein names, DNA, RNA, cell line and cell types, we leveraged ABNER (Settles 2005), a biomedical Named Entity Recognizer. We used OSCAR3[20] (Open Source Chemistry Analysis Routines) to identify chemical names and chemical structures. To annotate disease names, symptoms and causes, we used a subset of the Medical Subject Heading (MeSH) dataset. To extract event types and relations we built an extraction system that uses dependency parser and CRF based classifier.

3.3 MANAGEMENT OF XML DATA

3.3.1 Sedna XML database

As part of the BioEve system version 0.1, we used Sedna (Sedna n.d.) XML database which is an open source native XML database and comes with full-text search integrated with XQuery support. The native XML databases (NXDBs) store XML documents according to graph logical structure, in which the nodes represent elements and attributes and the edges define the element/sub-element and element/attribute relationships. These systems implement many characteristics that are common to traditional databases, such as storage, indexing, query processing, transactions and replication. The flexibility in representing XML data makes it difficult to typify, store, and process such documents.

[20] http://sourceforge.net/projects/oscar3-chem/

The management of XML data is complex. This is due, mainly, to the following characteristics (i) data model - XML documents are represented by a graph based data model, which increases the complexity of its structure (ii) heterogeneity – a XML document may have a sub-element completely absent or repeated several times. Regarding query processing, the XML model does not have a formal algebra yet. The W3C has developed formal semantics to the XPath and XQuery languages. However, it is complex, making it difficult to perform decomposing operations. Most of the native XML databases tend to choke for larger data and same is the performance issue experienced by us with Sedna. The challenges faced with Sedna are:

1. XQueries tend to become more complex as the data density increases. Example xquery used for Sedna is as below:

```
declare ordering unordered;
for $y in index-scan('pmid-by-idtext-a1files', "1800446", 'EQ')
let $names := $y//data(@EntityName)
let $types := $y//data(@EntityType)
where $names = 'benzene'
return
(
if ($types = ('PROTEIN','RNA','CELL LINE', 'CELL TYPE'))
  then <i>Protein/Genes: {$names}</i>
else (),
if ($types = 'DISEASE')
   then <i>Disease: {$names}</i>
else (),
if ($types = ('DRUG', 'CHEMICAL'))
```

then Drug/Chemical: {$names}

else ()

)

2. Frequent out of memory errors encountered.
3. Slow indexing of large xml files and slow xquery responses.
4. Doesn't support all the xquery-functions yet and XSLT support is limited.
5. For our data size (approximately 20 million abstracts from MedLine), Sedna didn't scale well with poorer xquery response time.

   Hence, for indexing full Medline data, we needed a better performing system with added features to make the query, search and response quality better while being scalable, highly available, easy-to-maintain search solution that doesn't cost a fortune to install. The solution we found is Solr[21].

3.3.2 Solr Indexing and Faceting Layer

Solr, an open source enterprise search platform suited our requirements. It comes with features of powerful full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling. Along with these features, Solr is highly scalable, providing distributed search and index replication making it one of the most used platforms by many of the world's largest internet sites (e.g. NASA, AT&T, Apple Inc., Disney etc) for the search and navigation features.

Apache Solr is written in Java and runs as a standalone full-text search server within a servlet container such as Tomcat. It uses the Lucene (Apache Software Foundation."Apache Lucene–Overview" n.d.) Java search library at its core for full-text indexing and search, Its REST-like HTTP/XML and JSON APIs makes it easy to use from virtually any programming language. The XML documents are indexed and on Solr

[21] http://lucene.apache.org/solr/

using HTTP POST and GET requests to the Solr Web application deployed in a servlet container. The default Solr response format is XML and we used it for faceted classification (Denton 2003) and navigation.

The annotated and tagged Medline abstract information is converted into XML format for Solr indexing. In order to facilitate indexing and faceting over the extracted semi-structured text articles, both web interface layer and faceting layer needs to share a common schema. A sample of shared schema used for BioEve is shown below:

```
<field name="pmid" type="string" indexed="true" stored="true" required="true" />
<field name="text" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="title" type="text" indexed="true" stored="true"/>
<field name="PubYear" type="text" indexed="true" stored="true"/>
<field name="gene" type="string" indexed="true" stored="true" multiValued="true" />
<field name="cell_type" type="string" indexed="true" stored="true" multiValued="true"
/>
<field name="cell_line" type="string" indexed="true" stored="true" multiValued="true"
/>
<field name="drug" type="string" indexed="true" stored="true" multiValued="true" />
<field name="rna" type="string" indexed="true" stored="true" multiValued="true" />
<field name="dna" type="string" indexed="true" stored="true" multiValued="true" />
<field name="disease" type="string" indexed="true" stored="true" multiValued="true" />
```

The Solr input xml file has the format as below:

```
<? xml version='1.0' encoding='UTF-8’?>
<add>
<doc>
<field name="pmid">6086383</field>
<field name="text"> Levels of collagenolytic activity produced by circulating polymorphonuclear leucocytes (PMN) of patients exposed to asbestos and patients with asbestosis were found to be similar to those of normal controls.</field>
<field name="cell_type">circulating polymorphonuclear leucocytes</field>
<field name="cell_type">circulating polymorphonuclear leucocytes</field>
<field name="cell_type">PMN</field><field name="drug">leucocytes</field>
<field name="drug">leucocytes</field><field name="drug">PMN</field>
<field name="disease"> asbestosis </field><field name="disease"> asbestosis </field>
<field name="Gene_expression">produced</field>
<field name="Negative_regulation">normal</field>
</doc>
</add>
```

Table 3.1 Solr input (XML) file sample

There are many other off the shelf systems are available such as in academia; Flamenco project[22] (from University of California Berkeley) and mspace[23] (University of Southampton) and in enterprise area; Apache Solr[24] and Endeca[25] which enables faceted classification and navigation over these facets/fields.

[22] http://flamenco.berkeley.edu/

[23] http://mspace.fm/

[24] http://lucene.apache.org/solr/

3.3 WEB INTERFACE

Web services are typically application programming interfaces (API) or web APIs that are accessed via Hypertext Transfer Protocol and executed on a remote system hosting the requested services. A typical web service web API is typically a defined set of Hypertext Transfer Protocol (HTTP) request messages along with a definition of the structure of response messages, usually expressed in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format[26]. Having faceting engine as a web service has many advantages as it abstracts interface design and implementation from faceting layer and also providing web APIs makes it easier for the user interface designer/developer.

We implemented BioEve user interface using AJAX, JavaScript and JSON to provide rich dynamic experience. Chapter 6 covers the user guidelines on how to use system's web-interface in more details. Also provided is a sample BioEve query/response using web API in section 6.2. The web API also allows bulk import of data with output either in XML or JSON format.

In terms of functional modules, the BioEve system consists of three main functional modules (S. P. Syed Toufeeq Ahmed 2010) as shown in Figure 1.1:

1. Extraction and Annotation Engine — To tag different gene/protein names, DNA, RNA, cell line and cell types, we leveraged ABNER (Settles 2005), a biomedical Named Entity Recognizer. We used OSCAR35 (Open Source Chemistry Analysis Routines) to identify chemical names and chemical structures. To annotate disease names, symptoms and causes, we used a subset of the Medical Subject Heading (MeSH) dataset.

[25] http://www.endeca.com/

[26] http://en.wikipedia.org/wiki/JSON, http://wiki.apache.org/solr/SolJSON

2. Semantic Faceted Search Server — Extraction engine pipeline is connected to a faceted search server with input as XML tagged abstracts (entities like genes, diseases and relationships). We used the Apache Solr library for faceted search (Tunkelang 2009), which also provides an enterprise quality full-text search.
3. BioEve Search Interface—The BioEve interface provides three features for cognitive search and navigation. The interface presents a number of entities types (on the left

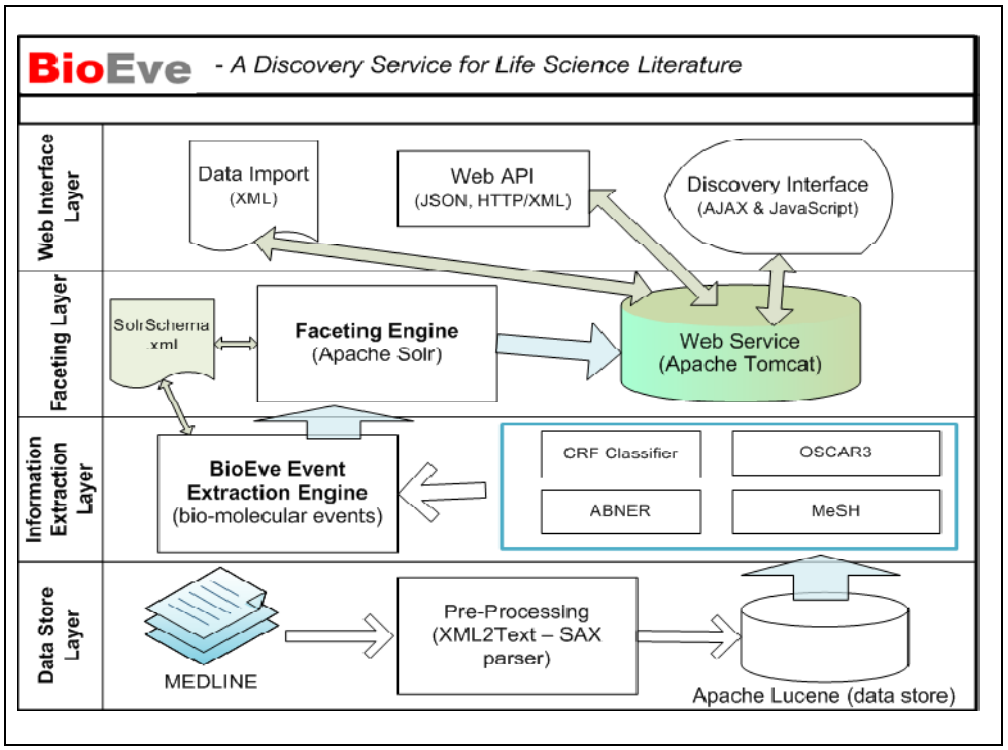

Figure 3.1: Layered Architecture of BioEve

panel) along with the specific instances/values, from previous search results, and the current query. Users can choose any of the highlighted values of these entity types to interactively refine the query (add new values/remove any value from the list with just one click) and thereby drill down to the relevant articles quickly without actually reading the entire abstracts. Users can easily remove any of the previous search terms, thus widening the

current search. The interface, implemented with the Web 2.0 technologies AJAX[27] and JavaScript, provides a rich dynamic feel and interactive usability without the need to reload the entire page after each refinement. The web interface runs on an Apache Tomcat server.

[27] http://github.com/evolvingweb/ajax-solr

CHAPTER 4

BIOMEDICAL NAMED ENTITY RECOGNITION

Biomedical literature contains a rich set of biomedical entities providing key information to access the knowledge. A biomedical named entity is a word or sequence of words that can be classified as name or biomedical term, such as gene, disease, protein, DNA, RNA, etc. The process of biomedical named entity recognition involves identifying and semantically classifying these terms into their correct categories. We applied ABNER (Settles 2005), a Biomedical Named Entity Recognizer, an open source software tool for molecular biology text mining , to tag different gene types including protein names, DNA, RNA, cell line and cell types. We used OSCAR3 (Oscar3 n.d.) (Open Source Chemistry Analysis Routines) to identify chemical names and chemical structures. To capture an ontology relationship, we used Unified Medical Language System (UMLS) Mesh database. These entities tagging operations are explained in more details below:

4.1 GENE TAGGING

To being with 2 million Medline abstracts, we used ABNER (Settles 2005) for the purpose tagging different gene names like proteins, cell line, DNA etc. ABNER, at its core, is a machine learning system using conditional random fields (CRFs) which employs a set of orthographic and semantic features. It has an overall *F* Measure of 70 in terms of recognizing occurrences of proteins, DNA, RNA, cell lines and cell types. However, the performance of ABNER is not very impressive. The following table summarizes the precision and recall for various entities learned by ABNER[28]:

[28] http://pages.cs.wisc.edu/~bsettles/abner/

Entity	Recall	Precision	F1	(S-F1)
Protein	77.8	68.1	72.6	(84.9)
DNA	63.1	67.2	65.1	(76.1)
RNA	61.9	61.3	61.6	(78.5)
Cell Line	58.2	53.9	56.0	(68.2)
Cell Type	65.6	79.8	72.0	(82.1)
Overall	72.0	69.1	70.5	(82.0)

Table 4.1NLPBA model. Five entities trained on 18,546 sentences, evaluated on 3,856

Entity	Recall	Precision	F1	(S-F1)
Protein	65.9	74.5	69.9	(83.7)

Table 4.2 BioCreative model. One entity (subsuming genes and gene products) trained on 7,500 sentences, evaluated on 2,500.

ABNER classifier for gene names is coupled with a disease dictionary created from MeSH to train Mallet based CRF classifier and OSCAR classifier for chemical names. This system helps in identifying the semantic relations in biomedical text in terms of *Gene−Disease* relations and *Disease−Treatment* relations as well as discovers hidden relations as explained in section 2.4. However, ABNER doesn't perform that well for the larger dataset of over 20 million abstracts available on PubMed at present. Hence for this big data, we have applied the dictionary based approach for tagging and annotation using HUGO (Tina A. Eyre 2006).

The HUGO[29] Gene Nomenclature Committee (HGNC) maintains a database of unique and approved human gene names and symbols. Current estimates predict the total number of protein coding human genes as 20□000–25□000, and over 18□000 of these now have been assigned HGNC approved nomenclature. The custom data download feature enables the users to download the HGNC data in both plain text and HTML format. Another important functionality with the custom download feature is that the

[29] http://www.genenames.org/

results are generated dynamically hence are up-to-date whenever the user returns to the saved URL. The performance of entity tagging process for over 20 million abstracts using this plain text file for dictionary based approach is very fast compared to ABNER in terms of execution hours.

4.2 DISEASE TAGGING

We used the Unified Medical Language System (UMLS) Mesh database to capture ontologies for entities present in abstract text. There are 49,712 entries, one for each tree number in 2009 MeSH, which contains 25,184 main headings 4. For tagging 2 million Pubmed abstracts more efficiently, an inverted index of Pubmed abstracts was created for each valid MeSH entry. It could also be used to identify broad areas which may interest the user, based on what the user searched. Also, a subset of these annotations was filtered out to annotate disease names, symptoms and causes.

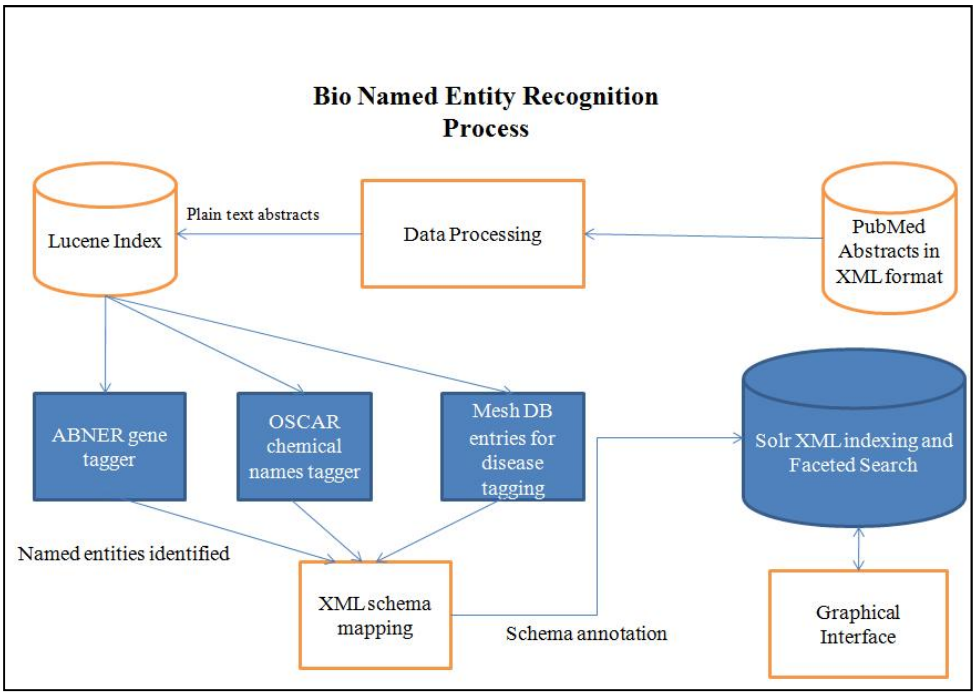

Figure 4.1 Bio Named entity recognition process

4.3 CHEMICAL/DRUG TAGGING

OSCAR (Open Source Chemistry Analysis Routines) is open source software for the semantic annotation of chemistry papers. This library uses various sources including hand - annotated texts, dictionary based on chemical names and structures. It not only recognizes chemical names, adjectives and processes, but is able to link them into their meaning using an ontology — a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant relationships. By using such a system, the researcher is freed from having to hunt for every permutation of a specific word, because OSCAR automatically links the word with its alternatives. It also can enrich the text-search by providing further information about the terms it identifies, such as chemical properties and molecular structure.

CHAPTER 5

ANNOTATION DESIGN

Traditional keyword search is based on syntactic matching, where documents returned contain the terms searched either partially or fully. However, there could be terms which are synonymous or semantically same as keywords searched for. Documents which have the term ”conjunctivitis” could be useful for the user, even if these documents do not contain the term ”eye diseases”. Biomedical Named Entity Recognition can be thought of as a sequence segmentation problem: each word is a token in a sequence to be assigned a label (Settles 2004). Inherently this tagging process captures entity type relationships, where the assigned label is a semantic parent class of the tagged entity. We captured two types of annotations; one which identifies biomedical entities and the other records event trigger phrases.

5.1 BIOMEDICAL ENTITY ANNOTATION[30]

This section describes in brief the tools and sources used by the Bio-Entity Tagging module.

- ABNER (Settles 2005): This tool is a Biomedical Named Entity Recognizer) is an open source software tool for molecular biology text mining. At its core is a machine learning system using conditional random fields with a variety of orthographic and contextual features (Settles 2005). We use ABNER in this work to tag protein names, DNA, RNA, cell line and cell types.
- OSCAR3 (Oscar3 n.d.): OSCAR3 (Open Source Chemistry Analysis Routines) is open source software for the semantic annotation of chemistry papers. This library uses various sources including hand - annotated texts, dictionary based on chemical

[30] This work is contributed by Radhika Nair

names and structures. Maximum Entropy Markov Model (MEMM) and N - gram based classifiers are some among the ones used for entity recognition.

- Ontology Tagging: We used the Unified Medical Language System (UMLS)[31] Mesh database to capture ontologies for entities present in abstract text. There are 49,712 entries, one for each tree number in 2009 MeSH, which contains 25,184 main headings[32]. For tagging 2 million PubMed abstracts more efficiently, an inverted index of PubMed abstracts was created for each valid MeSH entry. It could also be used to identify broad areas which may interest the user, based on what the user searched. Also, a subset of these annotations was filtered out to annotate disease names, symptoms and causes.

  Table 5.1 5 is an example of the bio - entity annotation. *Entity Type* records entity type information for biomedical entities, which can be one of the 7 predefined types, namely PROTEIN/GENE, DNA, RNA, CELL LINE, CELL TYPE, CHEMICAL/DRUG and DISEASE. The type CHEMICAL encompasses the drug category. Entities tagged as disease include disease names and also symptoms and causes for diseases. These annotations are referred to as *a*1 annotations.

ID	Entity Type	Start	End	Entity Name
T1	PROTEIN	290	307	steroid receptors

Table 5.1 Bioentity Annotations

Ontology annotations have the format as shown in Figure 5.2. Here, the entity type captures the semantic ontology valid MeSH entities in the abstract. The ontology representation is an ordered representation, where the labels represent parent - child relationship going from left to right. In the example shown below, *escherichia* is a

[31] http://www.nlm.nih.gov/research/umls/

[32] http://www.nlm.nih.gov/mesh/2009/download/mtr abt.html

specialized form of *enterobacteriaceae*, whereas *enterobacteriaceae* is a kind of *gammaproteobacteria* and so on.

ID	Ontology	Start	End	Entity Name
T8	[bacteria, proteobacteria, gammaproteobacteria, enterobacteriaceae]	471	484	escherichia

Table 5.2 Ontology Annotation

5.2 ANNOTATION FORMAT AND XML SCHEMA MAPPING

Our intent was to identify biomedical named entities and their semantic ISA relationships and Ontology relationships, events and their classification and convert it into an XML schema. This information is later uploaded to Solr as XML formatted document. Instead of tagging the actual plain text, we create a list of annotations for each PUBMED abstract. Each annotation (R. N. Syed Toufeeq Ahmed 2009) has a unique Annotation ID, annotation type, its start and end position in the abstract and the actual entity name.

5.3 XML SCHEMA[33]

Following a modular approach, we created four XML Schema designs. This section describes XML tree structure for each type of schema. PubMed ID is captured in all schemas to aid in cross references across tables.

Document Model

This schema captures the PubMed ID and the abstract text (includes the title as well). The XML tree structure for this schema is shown in Figure 5.1. PMID element stores unique PubMed ID assigned to each abstract. Abstract title and text are separated stored as attributes.

[33] This work is partly contributed by Radhika Nair

```
< events >
< PMID idtext=’8562886’ > < / PMID>
< AbstractTitle text=’The effect of Toremifene ··· ’><AbstractTitle>
<AbstractText text=’ Toremifene exerts multiple ···’> < /AbstractText>
< /events>
```

Figure 5.1. Document Model (a0 XML Schema)

Bioentity Model

This schema transforms biomedical entity annotations in to XML format as shown in Figure 5.2. The attribute *ID* records the entity ID, *Position* captures the start and end positions of the abstract in an ordered fashion. *Entity Type* records entity type information, which can be one of the 7 predefined types as discussed in section *Biomedical Entity Annotations*. Entity name is duplicated under *EntityName* as well as element value to allow flexibility in querying and as a option for any further changes to the format.

```
< events >
< PMID idtext=’8562886’ > < / PMID>
<AnnotationId ID=’T0’ Position=’60 83’ EntityType=’CELL TYPE’
EntityName=’human mononuclear cells’> human mononuclear cells
< /AnnotationId>
< /events>
```

Figure 5.2. Bioentity Model (a1 XML Schema)

Trigger Word Model

This schema currently does not incorporate complete event extraction including participants. Only trigger words are identified and mapped to schema shown in Figure

5.3. The elements and attributes are similar to a1 schema. *EntityType* attribute can take one of the predefined seven class labels.

Ontology Model

The only difference in this schema with respect to other schemas is that now the *EntityName* stores the ontology information in an ordered fashion. Figure 5.4 shows the a3 schema format. Possibly for the next version of this project, complete event annotations may be available to identify participants for events as well. These annotations would form a part of the

```
< events >
< PMID idtext=’8562886’ > < / PMID>
< TriggerAnnotationId text=’T8’ Position=’324 334’
EntityType=’Gene expression’ EntityName=’expression’> expression
< /TriggerAnnotationId>
< /events>
```

Figure 5.3. Trigger Word Model (a2 XML Schema)

```
< events >
< PMID idtext=’8562886’ > < / PMID>
<AnnotationId text=’T1’ Position=’210 220’ EntityType=’study
characteristics’ EntityName=’in vitro’> in vitro
< / AnnotationId >
< /events>
```

Figure 5.4. Ontology Model (a3 XML Schema)

a2 schema. Complete set of annotations for a sample abstract along with their corresponding XML Schemas are illustrated in Table 3.1.

Multi - Label Model

This schema captures multiple events per sentence. It is slightly different from other schemas in that, it does not store the actual sentence for which labels are stored. For each sentence the most likely events types present in that sentence is stored along with their probabilities.

```
< events >
< PMID idtext='8562886' > < / PMID>
<AnnotationId text='T8' Position='120 180' >
<EventType = "Gene expression" prob = "0.6"> < /EventType>
<EventType = "Phosphorylation" prob = "0.3"> < /EventType>
< / AnnotationId >
< /events>
```

Figure 5.5. Multi-Label Model (a4 XML Schema)

CHAPTER 6

BIOEVE: WEB-BASED USER INTERFACE

6.1 USER INTERFACE WALKTHROUGH

Interface is divided into left panel and right panel, see figure 6.1.

Left panel– offers suggestions and insights (based on co-occurrence frequency with the query terms) for different entities types, such as Genes, Diseases, Drugs/chemicals, Cell Lines, Cell types etc.

_Top of the left panel shows users current query terms and navigation so far (see “You Selected”). Here user can also de-select any of the previously selected entities or even all of them by single click on “remove all”. By de-selecting any entities, user is essentially expanding the search and the results in the right panel are refreshed on the fly to remaining query entities to offer a rich dynamic navigation experience.

_ Search box on left panel offers query entity suggestion (in light blue) as user types. These are frequently occurring entity names.

_ Below Search box, we have navigation categories (Genes, Diseases, Drugs/chemicals, Cell Lines, Cell types etc), user can click on any of the entity names (in light blue) to refine the search, by clicking user adds that entity name to the search and the results on the right panel are refreshed on the fly to reflect newly added query refinement.

_ User can add or remove number of refinements to the current search query until she/he reaches the desired results set (shown in the right panel).

Right Panel– shows user’s current search results and is automatically refreshed based on user’s refinement and navigation choices on the left panel.

_ Abstracts results on this panel show “title” of the abstract (in light red), full abstract text (in black, if abstract is available).

_ Below the full abstract text, the list of entities mentioned in that abstracts (in light blue) are shown. These entities names are click-able and will start a new search for that entity name, with just one click.

_ We also provide a direct link to the abstract page on PubMed.gov incase use wants to access additional information like authors and their affiliation.

_ Additionally we also provide a direct link to full text of the article from PubMed Central (PMC). Note: Only around 7% of the abstracts have full article text in PubMed Central, so most of the links to PMC will lead to page not found on PubMed central server.

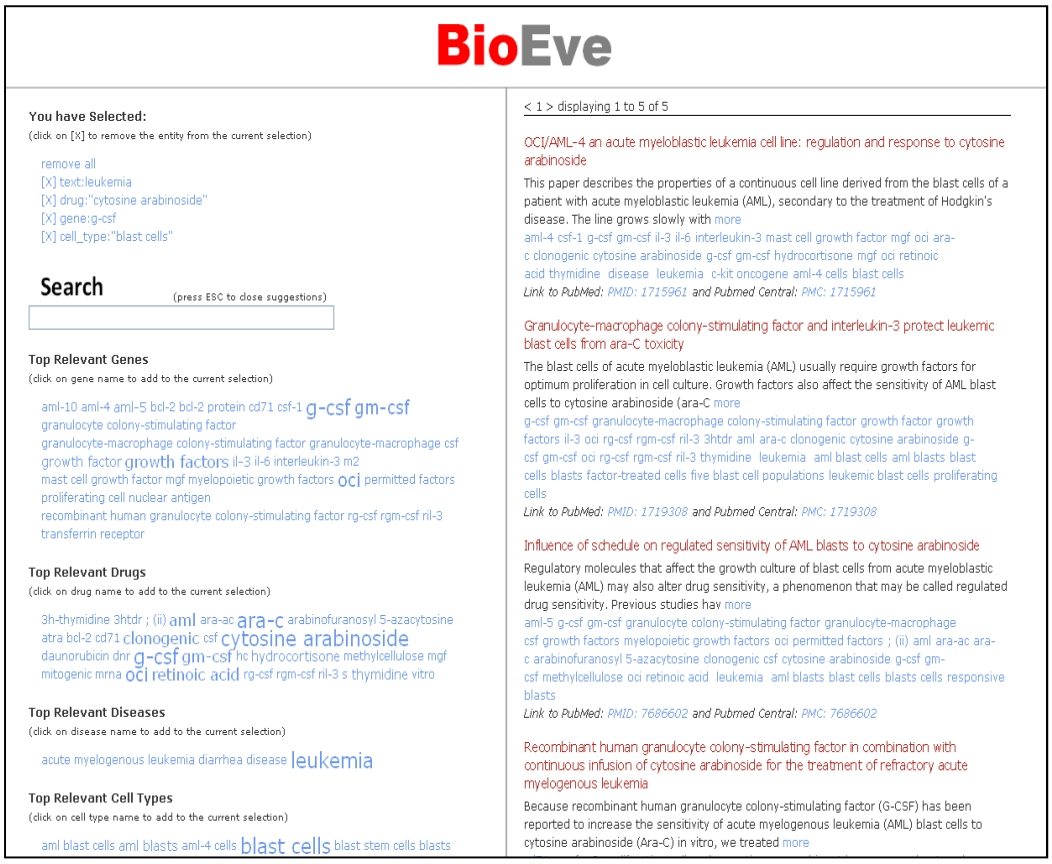

Figure 6.1 Screenshot of BioEve (version 1.0) user interface with the left panel and right panel view.

6.2 BIOEVE SAMPLE QUERY

BioEve facilitates the “discover more as you search” paradigm by making the sought information available with just a few clicks. It helps in facilitating what Swanson’s introduction of ‘transitive’ relations had demonstrated. For instance he fetched the most relevant literature reports about *fish oil* causing *reduction in blood viscosity* and *decrease in platelet aggregability*. He also identified another set of documents mentioning these symptoms as characteristics of *Raynaud’s syndrome*. Thus, based on these two findings he established the *hitherto unknown connection* which is *fish oil* can treat *Raynaud’s syndrome* thus revealing yet undiscovered relationships between different biomedical concepts found in the literature. BioEve aims to facilitate such findings through user friendly web-interface which shows such findings and helps in discovering new facts with just few clicks as shown in few steps below (and figure 6.2):

(a) - Let us start our search with query “Benzene”. BioEve’s auto-complete feature pulls out the name from its indexed terms while typing. The search results in 1986 articles, which we are now going to filter further to navigate to the information we are looking for.

(b) - Some helpful insights given by BioEve could be the Top Genes, in which “hemoglobin” is highlighted as a term frequently mention in the result set. Following this link narrows down the search to articles potentially discussing relationships between benzene and hemoglobin, resulting in 37 citations.

(c) - In Top Diseases, the user can see that the disease “leukemia” stands out, as it frequently occurs in the current result set. Leukemia is a blood cancer disease which is characterized by increasing while blood cell counts and falling hemoglobin count. Selecting “leukemia” refines the results further to 3.

(d) - If we scroll down to Top Cell Types, we find that “neutrophils” is highlighted. Clicking on “neutrophils” narrows the results to 2 publications, which can easily be browsed for details on the described association.

(e) - The highest ranked publication is titled “A hematology survey of workers at a styrene-butadiene synthetic rubber manufacturing plant”, see Figure 6.2. With a few clicks we discovered relevant articles that describe relationships between benzene, the disease leukemia, hemoglobin gene, and the neutrophils cell types.

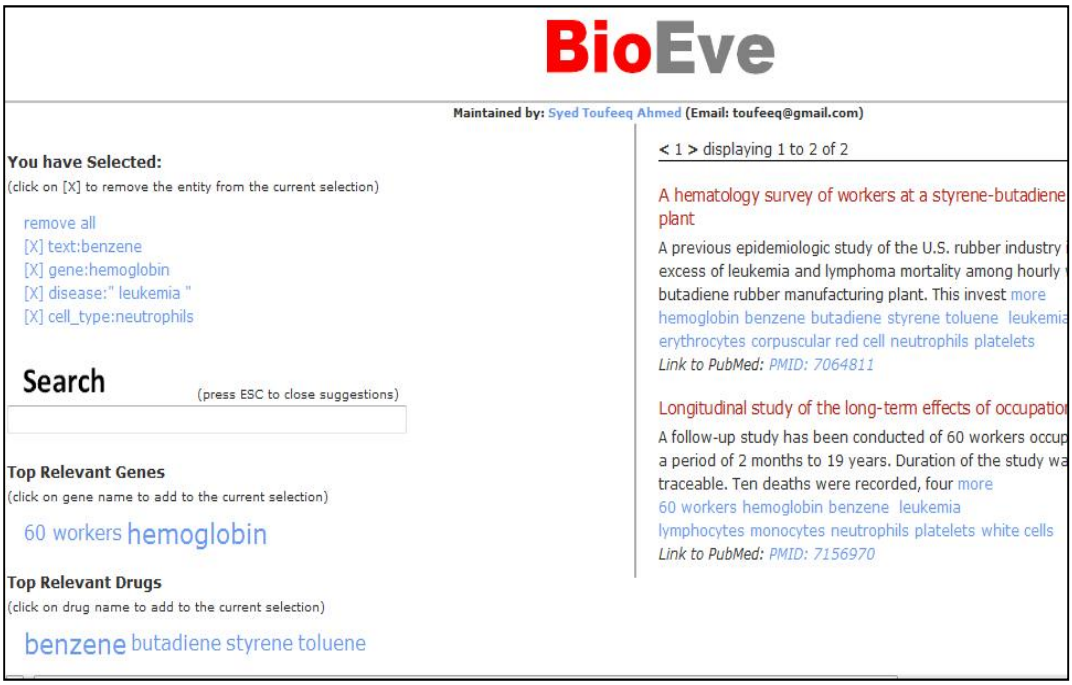

Figure 6.2 Screenshot of sample query final results

6.3 USER FEEDBACK AND COMMENTS

As part of performing system evaluation, we asked three life sciences researchers to evaluate BioEve search and their feedbacks are as below (paraphrased).

1. Dr. Fasahath Husain, Research Fellow, University of California, Berkeley, Email:fasahath@berkeley.edu

   I am a life science researcher and use PubMed extensively to search for research articles.

I also use Google Scholar but not nearly as much as Pubmed. Pubmed is a very popular search system because of historic reasons but the searching capability of Pubmed has lot to be desired.

Being a life science researcher I find many of syntax in Pubmed quite tedious. Something I don't want to learn or remember or even type in each time. On request of ww.bioeve.org developer, I tried the BioEve website. I am impressed by ease of its use. Although I did not get nearly all the data I was looking for, primarily because it is at preliminary stage, I think the search idea is headed in the right direction.

Strengths of BioEve:

1. Search could be a stepwise search, where we can refine our search until we get the precise content we are seeking.

2. If we require a certain alteration in the search results we can cancel a term from anywhere in the middle. This ability in search is very useful so that I do not have to restart my search.

3. Compared to Pubmed and Google Scholar the input for BioEve search is incredibly cognitive.

4. After the search, BioEve gives a direct link to Pubmed based on PMID; a big plus.

5. Unlike Google Scholar search, which is information overload, results from BioEve are quite refined.

Weaknesses and suggestions:

1. Pubmed and Google Scholar have integrated their search with local ISP. I have direct access to University library subscriptions when I use University ISP to search data using Pubmed (through library website) or Google Scholar. BioEve may have to work with Pubmed to make sure the search links are connected to local ISP.

2. Search based on date is something I miss. If I want to search for scientific development in a particular year, I can roughly find out using Google Scholar and Pubmed, but never exactly. I hope BioEve can introduce this option.

3. Perhaps because of its preliminary stage, I did not find nearly enough articles I looked for. There is extensive data about my topic, about 40,000 articles and I will be impressed if could search specifically within those articles.

When I have the confidence that BioEve is indexing all the data without missing any critical article, I will be compelled to use this search tool. I believe a finished product will be immensely useful and could become a popular tool for life science researchers.

2. Dr. Sukru Tuzmen, Adjunct Professor at Arizona State University, former Investigator and Head, Molecular Genetics Laboratory at Pharmaceutical Genomics Division, Translational Genomics Research Institute (TGen) through Skype

"You have a powerful search. Synchronize this with MEDLINE, add complete MEDLINE. Connect with more databases, OMIM, Entrez Gene ... You can get cell line database from ATCC.org"

"This is impressive! But you need to have all articles and new journals as they have fresh information, sometimes new published results cancel out old results."

3. Dr. Mostafa Afifi, Post-Doc Researcher, Faculty of Kinesiology, University of Calgary

As a researcher I found BioEve very useful as its a more focused research tool. I use pubmed on a regular basis and find its search tools difficult to use and at many times far from what I am looking for. BioEve on the other hand is much more user friendly and an easier user interface. I particularly like the idea of having larger fonts for the

more relevant terms highlighting what is researched more often. I applaud the creator for his effort!

CHAPTER 7

CONTRIBUTION AND FUTURE WORK

In this thesis report, we presented BioEve system, a discovery engine, with an intuitive and cognitive faceted search interface that enables discovery of important knowledge hidden in life sciences literature. BioEve (screenshot of version 1.0 is shown in figure 6.1, and is available at www.bioeve.org) demonstrates its potential and usability, but nonetheless we still have work to do to make it more complete and mainstream service.

Here are few next steps for us:

Processing and indexing of the entire MEDLINE collection:

BioEve currently has around 10% of the entire MEDLINE dataset. While processing and classifying the 1.9 million articles that are now available through a fast discovery interface, we saw that BioEve will scale well to 19 million articles and more.

Keeping BioEve up-to-date and complete:

We would like to add a synchronization module that will frequently check with MEDLINE for supplement articles as they are published; these will typically be in the range of 2500-4500 new articles per day. Frequent synchronization is necessary to keep BioEve abreast with MEDLINE collection and give users the access to the most recent articles.

Normalizing and grounding of entity names:

As the same gene/protein can be referred by various names and symbols (for example, the TRK-fused gene is also known as TF6; TRKT3; FLJ36137; TFG), a user searching for any of these names should find results mentioning any of the others. Removal of duplicates and clean-up of non-biomedical vocabulary that occurs in the entity tag clouds (like “conclusions”) will further improve navigation and search results.

Cross-referencing with biomedical databases:

We want to cross-reference terms indexed in BioEve with biological databases. For example, each occurrence of a gene could be linked to EntrezGene and OMIM; cell lines can be linked and enriched with ATCC.org's cell line database; we want to cross-reference disease names with UMLS and MeSH to provide access to ontological.

REFERENCES

Andreas Doms, Michael Schroeder. "GoPubMed: exploring PubMed with the Gene Ontology." Nucleic Acid Research, 2005: W783-786.

Andrei Mikheev, Steven Finch. "A workbench for finding structure in texts." Proceedings of the fifth conference on Applied natural language processing. Morristown, NJ: Association for Computational Linguistics, 1997. 372-379.

Apache Digester. Apache Software Foundation. “Digester–Commons“. http://jakarta.apache.org/commons/digester.

Apache Software Foundation.“Apache Lucene–Overview“. http://lucene.apache.org/java/docs/index.html.

Bénel A., Calabretto S., Iacovella A., Pinon J.-M. "Porphyry 2001: Semantics for scholarly publications retrieval." Proceedings of the thirteenth International Symposium on Methodologies for Intelligent Systems [ISMIS]. 2002. 351-361.

Blaschke C, Andrade MA, Ouzounis C, Valencia A. "Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions." Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology. AAAI Press, 1999. 60-67.

Brill, Eric. "A simple rule-based part-of-speech tagger." Applied Natural Language Conferences. Morristown, NJ: Association for Computational Linguistics, 1992. 152-155.

Christian Blaschke, Alfonso Valencia. "The Frame-Based Module of the SUISEKI Information Extraction System." IEEE Intelligent Systems, 2002: 14-20.

Cohen, WilliamW. "Infrastructure Components for Large-Scale Information Extraction." AAAI Press, 2003.

Denton, William. "How to Make a Faceted Classification and Put It On the Web." Nov 2003. http://www.miskatonic.org/library/facet-web-howto.html.

Fukuda K, Tamura A, Tsunoda T, Takagi T. "Toward information extraction: identifying protein names from biological papers." PSB, 1998: 705-716.

Hearst, Marti A. "Design recommendations for hierarchical faceted search interfaces." ACM SIGIR Workshop on Faceted Search. 2006.

—. User interfaces and visualization. Modern Information Retrieval, 1999.

Jonathan Koren, Yi Zhang, Xue Liu. "Personalized interactive faceted search." International World Wide Web Conference. NY: ACM, 2008. 477-486.

Lorraine Tanabe, W. John Wilbur. "Tagging gene and protein names in biomedical text." Proceedings of the ACL-02 workshop on Natural language processing in the biomedical

domain - Volume 3. Morristown, NJ: Association for Computational Linguistics, 2002. 9-13.

Mark Craven, Johan Kumlien. "Constructing Biological Knowledge Bases by Extracting Information from Text Sources." Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology. AAAI Press, 1999. 77-86.

Mary Elaine Califf, Raymond J. Mooney. "Relational learning of pattern-match rules for information extraction." American Association for Artificial Intelligence. Menlo Park, CA: American Association for Artificial Intelligence, 1999. 328-334.

Ono T, Hishigaki H, Tanigami A, Takagi T. "Automated extraction of information on protein–protein interactions from the biological literature." Bioinformatics, 2001: 155-161.

Oscar3. Prof. Peter Murray-Rust. “Oscar3“.http://sourceforge.net/projects/oscar3-chem/.

Philip J. Hayes, Steven P. Weinstein. "CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories." Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence. AAAI Press, 1990. 49-64.

Pyysalo, Sampo. A Dependency Parsing Approach to Biomedical Text Mining. Dissertation, TUCS, 2008.

Ryen W. White, Bill Kules,Steven M. Drucker and M.C. Schraefel. "Supporting Exploratory Search, Introduction." Communications of the ACM, 2006: 36-39.

Sedna. Sedna Native XML database. http://www.modis.ispras.ru/sedna/.

Settles, Burr. "ABNER An open source tool for automatically tagging genes, proteins and other entity names in text." Bioinformatics, 2005: 3191-3192.

—. "Biomedical named entity recognition using conditional random fields and rich feature sets." International Conference On Computational Linguistics. Morristown, NJ: Association for Computational Linguistics, 2004. 104-107.

Seymore K., McCallum A., Rosenfeld R. Learning hidden Markov model structure for information extraction. AAAI Technical Report, AAAI 99 Workshop on Machine Learning for Information Extraction, 1999, 37-42.

Shatkay, Hagit. Mining the Biomedical Literature : State of the Art, Challenges and Evaluation. Tutorial Program, Michigan: ISMB, 2005.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. "SemTag and seeker: bootstrapping the semantic web via automated semantic annotation." Proceedings of the 12th international conference on World Wide Web. NY: ACM, 2003. 178-186.

Syed Toufeeq Ahmed, Radhika Nair, Chintan Patel and Hasan Davulcu. "BioEve: Bio-Molecular Event Extraction from Text Using Semantic Classification and Dependency Parsing." BioNLP 2009. 2009. 99.

Syed Toufeeq Ahmed, Radhika Nair, Chintan Patel, Sheela P. Kanwar, Jrg Hakenberg, and Hasan Davulcu. "Semantic Classification and Dependency Parsing enabled Automated Bio-Molecular Event Extraction from Text." in proceedings of ACM International Conference On Bioinformatics and Computational Biology (ACM-BCB 2010). Niagara Falls, New York, USA.: ACM, 2010.

Syed Toufeeq Ahmed, Sheela P. Kanwar, Jrg Hakenberg, and Hasan Davulcu. "Bio- Eve: A Discovery Engine For Life Sciences Literature." in proceedings of 6th International Symposium on Bioinformatics Research and Applications (ISBRA10). Storrs, Connecticut, USA, 2010.

Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. "Automatic extraction of protein interactions from scientific abstracts." Proceedings of the Pacific Symposium on Biocomputing. 2000. 541-552.

Tina A. Eyre, Fabrice Ducluzeau, Tam P. Sneddon, Sue Povey, Elspeth A. Bruford and Michael J. Lush. "The HUGO Gene Nomenclature Database." Nucleic Acid Research, 2006: D319-21.

Tunkelang, Daniel. "Faceted Search." In Faceted Search (Synthesis Lectures on Information Concepts, Retrieval, and Services), by Daniel Tunkelang. Morgan & Claypool, 2009.

Udo Hahn, Martin Romacker, Stefan Schulz. "Creating Knowledge Repositories From Biomedical Reports: The MEDSYNDIKATE Text Mining System." Pacific Symposium on Biocomputing. 2002. 338-349.

Ulf Leser, Jorg Hakenberg. "What makes a gene name?Names entity recognition in the biomedical literature." Briefings in BioInformatics, 2005: 357-369.

Vineet Sinha, David R. Karger. "Magnet: Supporting Navigation in Semistructured Data Environments." Proceedings of the 2005 ACM SIGMOD international conference on Management of data. Baltimore, Maryland: ACM, 2005. 97-106.

Walter Daelemans, Sabine Buchholz , Jorn Veenstra. "Memory-Based Shallow Parsing." CoNLL. 1999. 53-60.