Text analytics for life science using the Unstructured Information Management Architecture

by R. Mack
S. Mukherjea
A. Soffer
N. Uramoto
E. Brown
A. Coden
J. Cooper
A. Inokuchi
B. Iyer
Y. Mass
H. Matsuzawa
L. V. Subramaniam

Biomedical text plays a fundamental role in knowledge discovery in life science, in both basic research (in the field of bioinformatics) and in industry sectors devoted to improving medical practice, drug development, and health care (such as medical informatics, clinical genomics, and other sectors). Several groups in the IBM Research Division are collaborating on the development of a prototype system for text analysis, search, and text-mining methods to support problem solving in life science. The system is called "BioTeKS" ("Biological Text Knowledge Services"), and it integrates research technologies from multiple IBM Research labs. BioTeKS is also the first major application of the UIMA (Unstructured Information Management Architecture) initiative also emerging from IBM Research. BioTeKS is intended to analyze biomedical text such as MEDLINE™ abstracts, medical records, and patents; text is analyzed by automatically identifying terms or names corresponding to key biomedical entities (e.g., "genes," "proteins," "compounds," or "drugs") and concepts or facts related to them. In this paper, we describe the value of text analysis in biomedical research, the development of the BioTeKS system, and applications which demonstrate its functions.

The large scale sequencing of the human genome has greatly increased our knowledge of the genetic basis of biological processes and accelerated the pace of research and development aimed at treating disease and enhancing the health and well-being of humans. However, these advances also result in increased complexity in understanding and applying biomedical research and data. There is consensus in the life-science (LS) industry and academic laboratories that managing the complexity of biological data and knowledge requires an integrative, information-based systems approach, in which computer technology must play an essential role. For a cogent analysis of this situation and the role of computational methods in life science, see References 1–3.

Key components of computational technology that are relevant to this effort include analyzing, searching, and mining biomedical text, and correlating the structured data derived from texts with data derived from biomedical experiments, transcribed medical records, and so on. This paper describes an IBM Research project to exploit and develop the text-analytical technology needed for managing, analyzing, and using biomedical text to solve problems in life science. We call the system *BioTeKS* for "Biological Text Knowledge Services." BioTeKS is also one of the first major systems implemented with the IBM Unstructured Information Management Architecture (UIMA), which is described later in this paper, in other papers in this issue,[4] and elsewhere.[5]

This paper begins by describing the role and value of text analysis in LS research and development, and how BioTeKS fits into the broad range of technologies needed to manage text content. It then focuses in detail on the BioTeKS system specifically, and how BioTeKS is being used to explore text analysis, text search, and text mining to support problem solving in life science.

The role of text analysis in life-science research and development

Text analysis is a key component in text-oriented unstructured information management (UIM). The general goal of text analysis in UIM is to transform unstructured text information into structured information, and to use this information to support higher-level processes of text search, mining, and discovery. (For comprehensive reviews of UIM, see References 4, 6, and 7.) Transforming unstructured text into structured information means transforming "chunks" of text into specific, discrete data objects categorized or labeled by one or more attributes, where the data objects are words, phrases, or larger text segments. The essence of what the BioTeKS system does is information extraction (IE) for life-science text. Examples of IE include identifying names of biomedical entities, like gene, protein, and disease names (which may be expressed in multiword phrases), and identifying more complex facts about and relations between entities, such as interactions between proteins, genes, and the functions associated with them, or the correlations between drug effects and disease indications. Several overviews of text IE exist, in general,[8,9] and specifically for life science,[10–12] and we assume readers have some familiarity with the basic technical issues in IE.

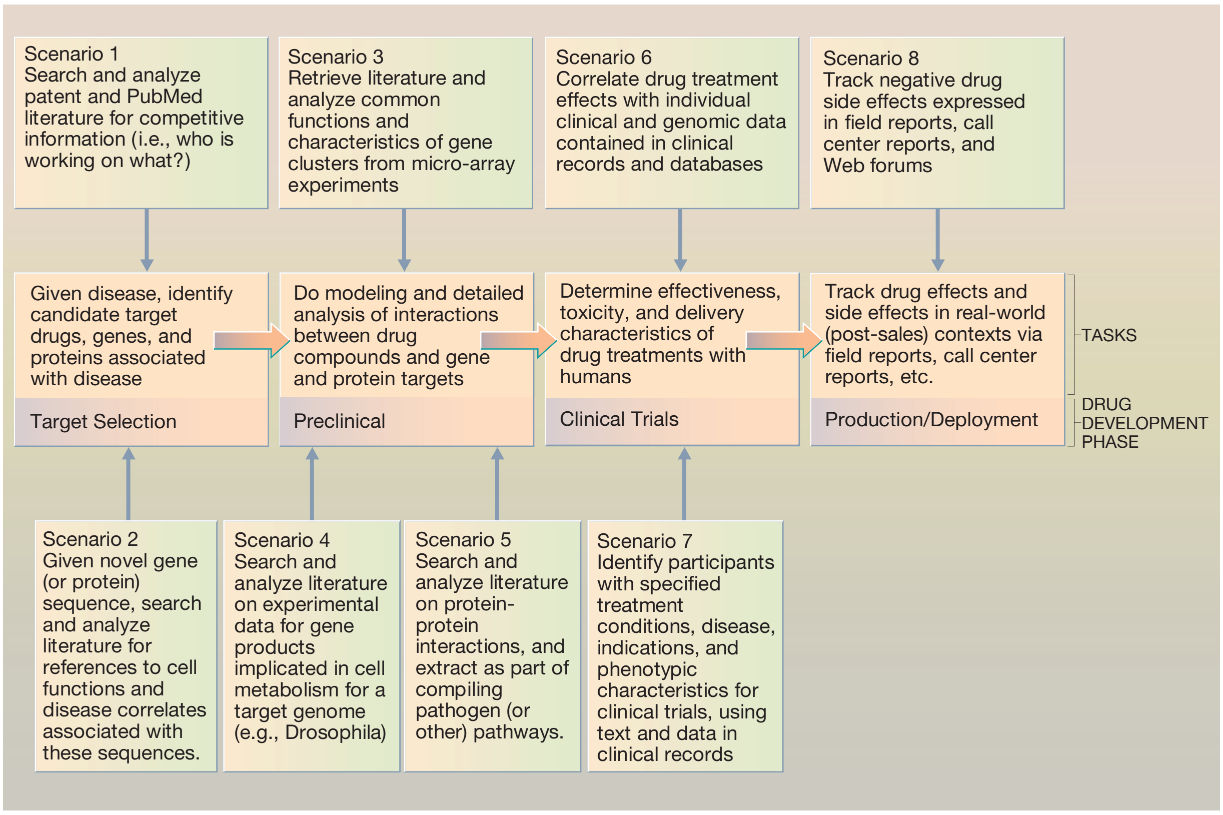
The business and research value of extracting structured text information is that it can be used to solve problems in key biomedical domains and increase productivity in research and development. Text analysis can enhance general knowledge management practices and tools, for example, by improving the effectiveness (i.e., precision) of searching for documents in large collections, and by organizing these collections into taxonomy groupings or topic clusters for easier browsing. More importantly, text analysis can support knowledge discovery in various domains. Papers on knowledge portals and text-mining research in IBM can be found in recent special issues of the *IBM Systems Journal*.[6,7,13,14]

Figure 1 provides examples of text analysis phases in life science in relation to four key phases of drug research and development (see Reference 3). In the "Target Selection" phase for a drug (scenario 1), for example, a researcher needs to search scientific and patent literature to find out what drugs or diseases other researchers or institutions are working on, and what is already known and patented in this field. Knowledge discovery can increase the speed (and hence the productivity) of a drug researcher finding a drug target, a competitor's patent activity, or a participant in a clinical trial. In the "Preclinical" phase, researchers may conduct experiments that provide indications of relevant gene responses to drugs or disease agents. Scenarios 2, 3, 4, and 5 all pertain to finding literature describing aspects of genes and proteins that can help researchers investigate hypotheses about the relevance of these genes or gene products to some drug, disease, or biological process of interest. For example, studies have shown that literature references to genes can improve the search for gene homologies that may be relevant to identifying functions of novel target genes[15] (scenario 2), validate molecular pathways,[16,17] and help interpret why a cluster of genes might react together under some experimental conditions[18] (scenario 3).

In later stages of drug development, the effectiveness of target drugs is evaluated in the "Clinical Trials" phase (scenario 6). Text analysis of medical records can provide information which can be combined with other kinds of more traditional structured data (e.g., individual genomic information), to assist in data mining of factors associated with positive or negative drug and treatment effects. Text mining can be used to find participants for a clinical trial (scenario 7), based on specific attributes of treatment, medical history, and so on. Finally, once a drug is in the marketplace, feedback about large-scale use under uncontrolled circumstances provides additional valuable feedback. Text mining can be used to analyze treatment effects as reported in various forums, field reports, and call center records (scenario 8).

In each of these scenarios, text analysis has the potential for reducing the time it takes researchers to find relevant documents and to find specific factual content within documents that can help researchers interpret experimental data, clinical record information, and business intelligence data contained in patents. As Ng and Wong put it, "The race to a new gene or drug is now increasingly dependent on how quickly a scientist can keep track of the voluminous information online to capture the relevant picture

Figure 1 Text–mining scenarios for four phases of drug development

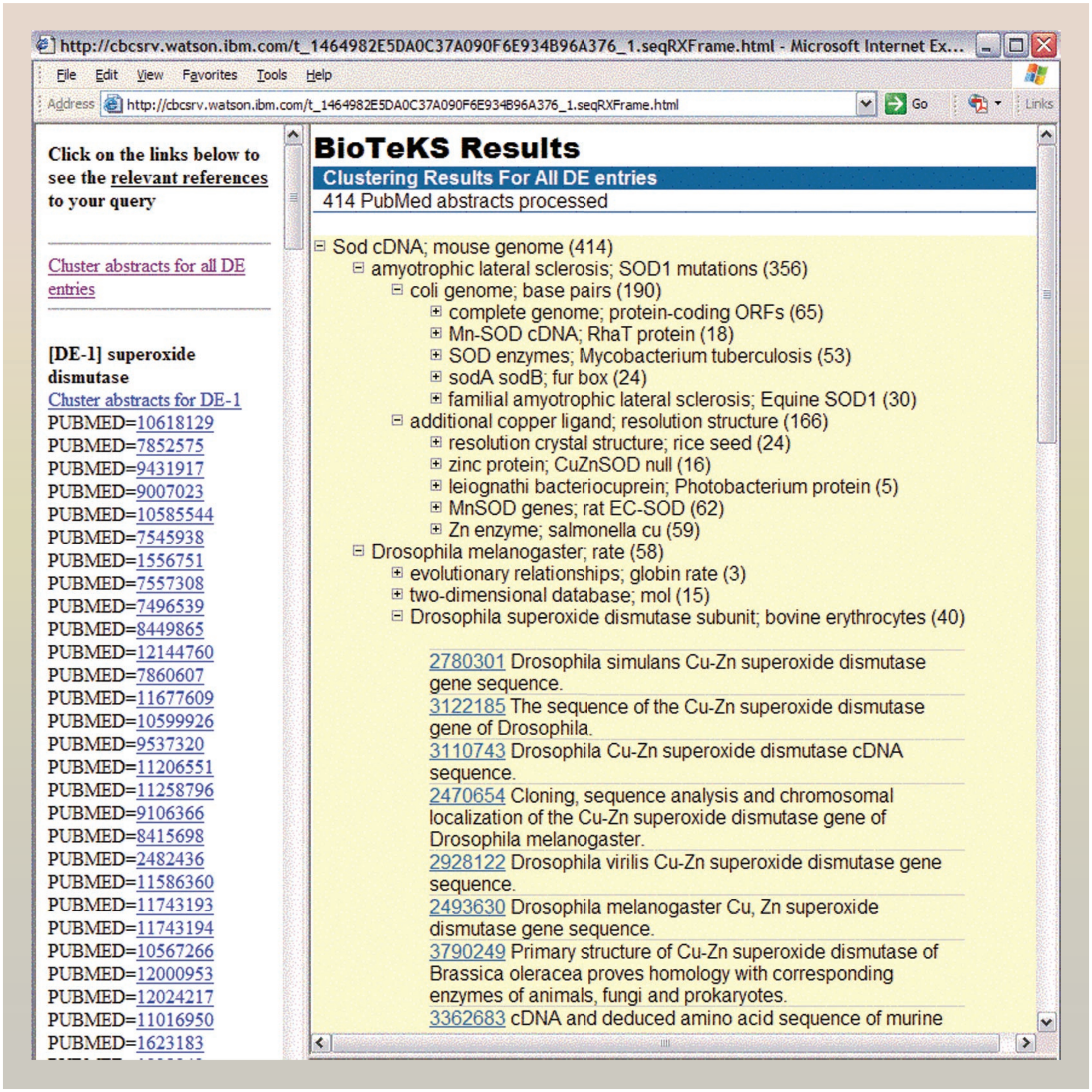

Scenario 1
Search and analyze patent and PubMed literature for competitive information (i.e., who is working on what?)
Scenario 3
Retrieve literature and analyze common functions and characteristics of gene clusters from micro-array experiments
Scenario 6
Correlate drug treatment effects with individual clinical and genomic data contained in clinical records and databases
Scenario 8
Track negative drug side effects expressed in field reports, call center reports, and Web forums
Given disease, identify candidate target drugs, genes, and proteins associated with disease
Target Selection
Do modeling and detailed analysis of interactions between drug compounds and gene and protein targets
Preclinical
Determine effectiveness, toxicity, and delivery characteristics of drug treatments with humans
Clinical Trials
Track drug effects and side effects in real-world (post-sales) contexts via field reports, call center reports, etc.
Production/Deployment
TASKS
DRUG DEVELOPMENT PHASE
Scenario 2
Given novel gene (or protein) sequence, search and analyze literature for references to cell functions and disease correlates associated with these sequences.
Scenario 4
Search and analyze literature on experimental data for gene products implicated in cell metabolism for a target genome (e.g., Drosophila)
Scenario 5
Search and analyze literature on protein-protein interactions, and extract as part of compiling pathogen (or other) pathways.
Scenario 7
Identify participants with specified treatment conditions, disease, indications, and phenotypic characteristics for clinical trials, using text and data in clinical records

(such as protein-protein interaction pathways) hidden within the latest research articles."[16]

These text-based search and mining tasks are typically functions available in end-user applications. BioTeKS has not yet been used to build all the applications implied in Figure 1. These applications tend to be highly customized for specific needs and practices of specific pharmaceutical organizations. BioTeKS has focused instead on the core text analysis methods needed to extract the text data that such applications would use as input for further higher-level processing; that is, indexing a text search engine or clustering search results, mining trends, or associations among terms expressing concepts of interest.

Document clustering using BioTeKS components can be found as a function in the publicly accessible Web-based Bio-Dictionary* tool[19] developed by the Computational Biology Center[20] at the IBM Watson Research Center. Bio-Dictionary uses pattern-matching algorithms to analyze protein sequences in relation to molecular-level subunits called "seqlets" (seqlets are to proteins roughly what words and phrases in a dictionary are to sentences that use them to express ideas). Much of the protein analysis has nothing to do with text, per se, but as the system searches for known and homologous proteins on public protein databases such as Swiss-Prot**,[21] it also retrieves and compiles MEDLINE** abstracts that describe facets of these proteins. These documents contain descriptions of potentially relevant functions and characteristics of the proteins under analysis. The document clustering results shown in Figure 2 are a subset of several hundred MEDLINE documents compiled by Bio-Dictionary for the sample protein sequence "SODUM-BORU," and the clusters are labeled with topically relevant keywords. These include phrases suggesting attributes and functions. Hierarchical document clustering with cluster labels helps

Figure 2 Hierarchical document clustering results


http://cbcsrv.watson.ibm.com/t_1464982E5DA0C37A090F6E934B96A376_1.seqRXFrame.html - Microsoft Internet Ex...
File Edit View Favorites Tools Help
Address http://cbcsrv.watson.ibm.com/t_1464982E5DA0C37A090F6E934B96A376_1.seqRXFrame.html
Go
Links

Click on the links below to see the relevant references to your query

Cluster abstracts for all DE entries

[DE-1] superoxide dismutase
Cluster abstracts for DE-1
PUBMED=10618129
PUBMED=7852575
PUBMED=9431917
PUBMED=9007023
PUBMED=10585544
PUBMED=7545938
PUBMED=1556751
PUBMED=7557308
PUBMED=7496539
PUBMED=8449865
PUBMED=12144760
PUBMED=7860607
PUBMED=11677609
PUBMED=10599926
PUBMED=9537320
PUBMED=11206551
PUBMED=11258796
PUBMED=9106366
PUBMED=8415698
PUBMED=2482436
PUBMED=11586360
PUBMED=11743193
PUBMED=11743194
PUBMED=10567266
PUBMED=12000953
PUBMED=12024217
PUBMED=11016950
PUBMED=1623183

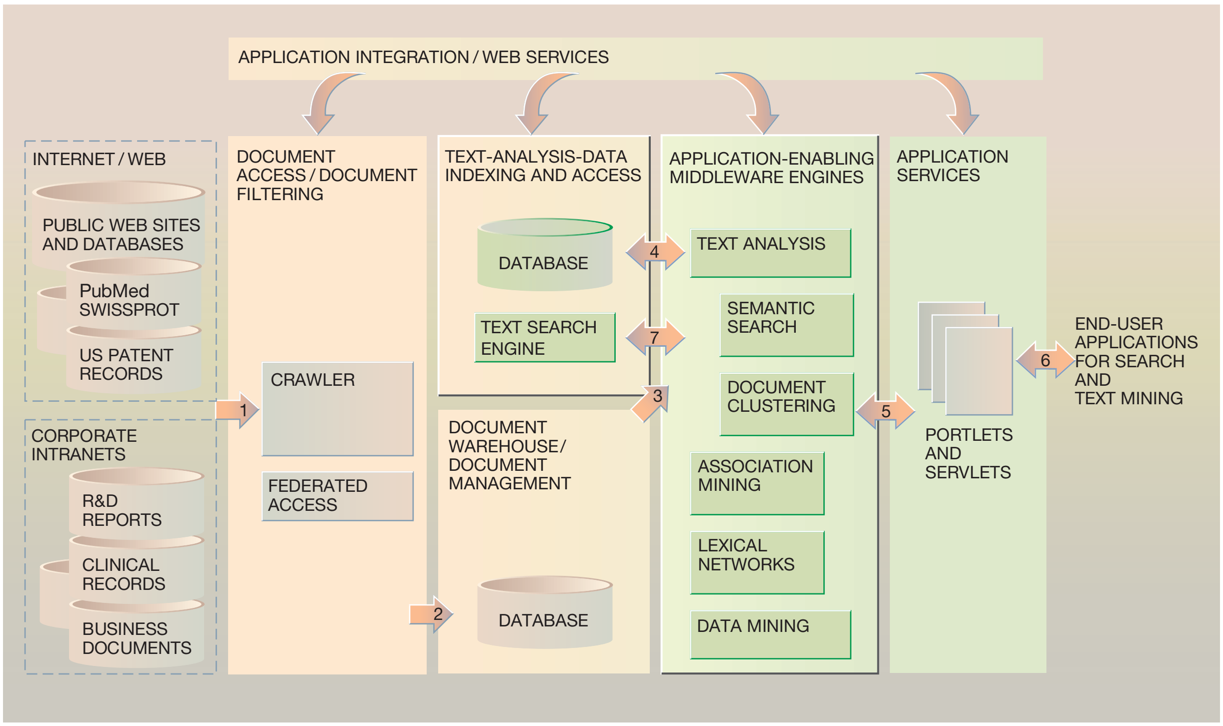
BioTeKS Results
Clustering Results For All DE entries
414 PubMed abstracts processed

- Sod cDNA; mouse genome (414)
  - amyotrophic lateral sclerosis; SOD1 mutations (356)
    - coli genome; base pairs (190)
      - complete genome; protein-coding ORFs (65)
      - Mn-SOD cDNA; RhaT protein (18)
      - SOD enzymes; Mycobacterium tuberculosis (53)
      - sodA sodB; fur box (24)
      - familial amyotrophic lateral sclerosis; Equine SOD1 (30)
    - additional copper ligand; resolution structure (166)
      - resolution crystal structure; rice seed (24)
      - zinc protein; CuZnSOD null (16)
      - leiognathi bacteriocuprein; Photobacterium protein (5)
      - MnSOD genes; rat EC-SOD (62)
      - Zn enzyme; salmonella cu (59)
  - Drosophila melanogaster; rate (58)
    - evolutionary relationships; globin rate (3)
    - two-dimensional database; mol (15)
    - Drosophila superoxide dismutase subunit; bovine erythrocytes (40)

2780301 Drosophila simulans Cu-Zn superoxide dismutase gene sequence.
3122185 The sequence of the Cu-Zn superoxide dismutase gene of Drosophila.
3110743 Drosophila Cu-Zn superoxide dismutase cDNA sequence.
2470654 Cloning, sequence analysis and chromosomal localization of the Cu-Zn superoxide dismutase gene of Drosophila melanogaster.
2928122 Drosophila virilis Cu-Zn superoxide dismutase gene sequence.
2493630 Drosophila melanogaster Cu, Zn superoxide dismutase gene sequence.
3790249 Primary structure of Cu-Zn superoxide dismutase of Brassica oleracea proves homology with corresponding enzymes of animals, fungi and prokaryotes.
3362683 cDNA and deduced amino acid sequence of murine

users browse document collections more easily than unordered lists of MEDLINE titles, especially when the cluster labels suggest topics for subsets of MEDLINE abstracts. The clustering engine, an example of BioTeKS application-enabling middleware, uses noun phrases in its clustering algorithm and for labeling each cluster. BioTeKS annotators (described later) extract these noun phrases and make them available to the clustering engine. Other BioTeKS components format the results of clustering as an XML (eXtensible Markup Language) file that can be presented to an end user as a dynamic (interactive) HTML (HyperText Markup Language) Web page, as shown in Figure 2.

Figure 3 Generic platform for managing unstructured (text) information



In this context, BioTeKS functions as a text-analysis middleware component of a larger system that starts with input documents obtained from some source (e.g., search processes in the Bio-Dictionary tool), and ends as an end-user application view (e.g., the Web page in the figure) that enables end users to do something useful with text documents, such as browse them by means of a hierarchy of labeled clusters. The labels suggest specific topics relevant to interpreting the characteristics of the original novel protein.

In the next section, we discuss the larger middleware technology context for BioTeKS, and then we devote the rest of the paper to describing BioTeKS text-analysis methods.

Technologies for managing text content

A variety of technologies are needed to support text-based knowledge management and discovery, from accessing and managing documents (unstructured information) from various sources to delivering analysis results to end user applications. Figure 3 shows a generic platform depicting this end-to-end sequence of technologies. BioTeKS focuses on the green components in the figure: text-analysis methods, indexing, and storage of text-analysis data. Application middleware engines access and further analyze this text data. Many products are available from IBM and other vendors to implement this platform (see the Life Sciences Framework[22] for a description of available IBM products and technologies).

Managing unstructured text information begins with accessing text information using crawling or federated search methods. Once text is accessed, it needs to be filtered or converted from a variety of formats (e.g., PDF [Portable Document Format], HTML, XML) into a standard form for further processing. The latter includes document parsing to extract text content ("blobs") and meta-data, and it often involves storing this information in local data warehouses within an organization. Collected text information is typically stored in diverse and heterogeneous repositories, including public and corporate Web sites and internal corporate databases (e.g., Lotus Notes, file systems). These basic functions are available in standard IBM products (e.g., IBM DB2* Information

Integrator for Content[23]) or other vendor products in the domain of content and document management.

Once the basics of text access and warehousing or storage are accomplished, text analysis of various kinds can be carried out at a finer granularity, to support a wide range of text searching and mining applications. This is where the BioTeKS system and UIMA come into play. The numbered arrows in Figure 3 show the high-level flow of information between these components. Accumulated documents flow in to the text analysis component (arrow 3). Text analysis processes in this component automatically extract text features of various kinds and annotate these features with linguistic and semantic information that associates meaning with text features (e.g., a string is a "gene" relative to some resource). Text analyses produce fine-grained, structured text data that typically need to be stored and indexed for use by higher-level applications. Structured data extracted from the document as meta-data (e.g., author, title, date, keyword annotations based on MeSH [Medical Subject Headings] codes, etc.) and extracted from text content (e.g., entity names such as genes, drugs, or diseases contained in MEDLINE abstracts) can be stored (arrow 4) in a relational database for access by upstream text-mining applications using SQL (structured query language) database queries. In addition, text content can be indexed in a text search engine, enabling text search based on many more terms indexed in the full document text content.

The application-enabling engines identified in Figure 3 provide access to text-analysis data for text-search and higher-level text-mining functions, including tools that allow researchers to control these analyses and visualize results. Arrow 7 in the figure indicates access to indexed information by application-level components, and arrow 5 indicates how end-user (Web) applications communicate user requests to these middleware application components. For example, the user may submit a text query in some query language, which is ultimately provided as input to the text search engine via the "semantic search" component in the figure. This component can also contain additional functions to enhance searching, for example, to help users express search intentions in query languages, or to iteratively refine searches by suggesting query terms based on users selecting result documents they judge to be relevant to their interests (i.e., "show me more like this"). Searching can also be enhanced by ranking result documents by their relevance to query terms, and by clustering or categorizing documents, making it easier for users to browse large collections of search result documents.

Text-mining functions provide additional ways of organizing and visualizing text, either at the document level or at the level of more specific text content. Text-mining methods include analyzing associations and trends between categories of entities, such as correlations between names of researchers and research topics, genes and gene products, drug and compound effects and disease indications, and so on. We discuss these and other text-mining applications in more depth later.

The technology platform indicated in Figure 3 includes tools for developing applications, including graphical user interfaces and Web services to manage the application logic supporting application clients. These services are used for controlling the integration and information flow among middleware components like databases and search engines and document-clustering engines. In the section, "Application prototypes and application-enabling engines," we discuss examples of such applications.

An overview of the BioTeKS system

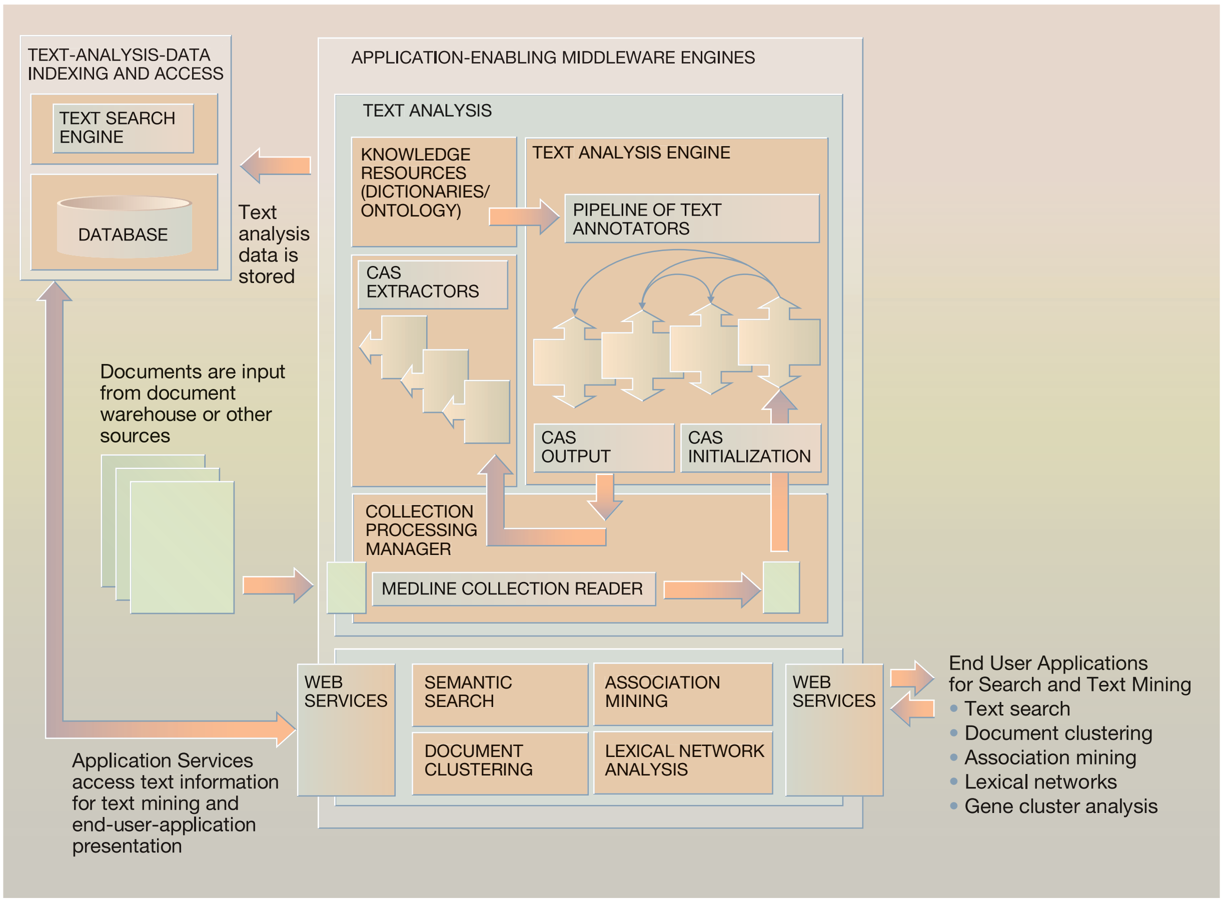
BioTeKS is a system focused on methods for analyzing or annotating biomedical text, including a scheme for storing and indexing text-analysis data generated from these methods and a suite of prototype application-enabling engines for supporting certain types of biomedical text-mining applications.

Figure 4 presents specific text-analysis components of BioTeKS which were shown in green in Figure 3. As shown, the text-analysis engine applies a sequence of text annotators to annotate input documents. Text analysis data is stored in a database and in a text search engine, and application-enabling engines access this text data and apply additional text-mining methods to the data.

BioTeKS functions in a larger context where documents have already been crawled or compiled into a collection, and a set of applications of interest might already exist (e.g., text search applications). BioTeKS focuses on the analysis and extraction of text information from documents in this collection to support text searching and mining applications.

BioTeKS uses the UIMA framework and tools from which it derives much of its value. The component

Figure 4 Overview of BioTeks system and UIMA


TEXT-ANALYSIS-DATA INDEXING AND ACCESS
TEXT SEARCH ENGINE
DATABASE
Text analysis data is stored
APPLICATION-ENABLING MIDDLEWARE ENGINES
TEXT ANALYSIS
KNOWLEDGE RESOURCES (DICTIONARIES/ ONTOLOGY)
TEXT ANALYSIS ENGINE
PIPELINE OF TEXT ANNOTATORS
CAS EXTRACTORS
CAS OUTPUT
CAS INITIALIZATION
COLLECTION PROCESSING MANAGER
MEDLINE COLLECTION READER
Documents are input from document warehouse or other sources
WEB SERVICES
SEMANTIC SEARCH
ASSOCIATION MINING
DOCUMENT CLUSTERING
LEXICAL NETWORK ANALYSIS
WEB SERVICES
Application Services access text information for text mining and end-user-application presentation
End User Applications for Search and Text Mining
• Text search
• Document clustering
• Association mining
• Lexical networks
• Gene cluster analysis

labeled "Text Analysis Engine" in Figure 4 also describes the essential elements of the UIMA implementation of BioTeKS. Other papers describe UIMA in depth, and we only highlight aspects of it relevant to describing BioTeKS.[4,5]

How UIMA improves text analysis processing. UIMA (and BioTeKS) address the challenges of analyzing text by standardizing and improving the process of developing, deploying, and integrating the operation of multiple text-analysis methods that operate on different levels of linguistic analysis.[5] Document text expresses ideas that involve all the complexity of human language expression. Analyzing the form and content of text is complicated because it entails integrating and coordinating analysis methods for multiple levels of analysis of linguistic structure, from character strings and word tokens to names and phrases, to syntactic elements including clauses and sentences, to still higher-level topic structures that might span multiple paragraphs within documents and even multiple documents. Each of these levels of information requires specialized analysis methods, all of which need to be orchestrated and integrated.

How BioTeKS improves text analysis processing for life science. BioTeKS in turn is focused on methods for text analysis of biomedical text, which is especially complicated scientific text. What these methods are, and how these methods improve the analysis of biomedical text, is the subject of the rest of this paper. The starting point, however, is to use UIMA as the framework to develop and orchestrate a collection of text-analysis methods that automatically identify names of biomedical entities (e.g., genes and

Table 1 Annotators used in BioTeKS

Annotator Name	Function in BioTeKS	Comments
LanguageWare linguistic engine	Used for tokenizing text into strings and sentences. Assigns generic lexical information to strings.	Tokenization based on regular expression rules and dictionary lookup. Dictionary lookup also provides generic lexical (lemma) information (e.g., part of speech). Dictionaries can be customized for specific categories of entities.
POS (part of speech) tagger	Assigns parts of speech to tokens, using context.	Assigns parts of speech to tokens based on statistical model of terms in context. Can also disambiguate POS annotations from prior annotators. POS tagger can be trained on manually annotated document corpus. Default training corpus is licensed from Language Data Consortium.[28]
FST (finite state transducer) with shallow parsing rules	Parses sentences into syntactic units (e.g., subject noun phrase).	General-purpose FST engine. Syntactic parsing rules customize the FST for shallow parsing. Shallow parser assumes POS tags of individual string tokens.
Dictionary Lookup	Assigns a lexical or semantic category to a string (e.g., "Trk A is a protein") and other lexical information	General-purpose dictionary-based pattern-matching lookup. Takes tokens as input or does its own tokenization. Includes reference to an authority (e.g., MeSH), canonical form, and synonyms. Multiple dictionaries for flexible entity extraction; e.g., MeSH terms, genes, proteins, and drugs. Includes functions for handling case, stemming, etc. Dictionaries developed as XML files.
BioAnnotator	Identifies biomedical terms and phrases and associates a UMLS identifier with them.	General-purpose dictionary-lookup method specialized for UMLS. Takes noun phrases as input and classifies them as "biological" if they match selected categories of UMLS terms or contain UMLS terms. Includes regular expression rules for disambiguating terms in context.
ChemFrag	Identifies complex tokens as organic chemical names.	Identifies organic chemical names using rules based on chemical fragment strings. Does not identify specific chemical name with respect to chemical name-resource.
DrugDosage	Identifies phrases containing drug name and a dosage qualifier.	Uses dictionary lookup for drug names, quantities, and dosage qualifiers and FST rules for identifying phrasal combinations of these elements.
Term Ontology Mapper	Adds taxonomy information to identified terms.	Dictionary lookup methods identify terms relative to ontology and dictionary resources, but in some cases, like MeSH (Medical Subject Headings), there is additional information for placing terms in a hierarchical taxonomy. This annotator fills in this additional semantic information.
Term Categorizer	Assigns a lexical and semantic category to a string (e.g., "congestive heart failure" is a disease symptom).	Term Categorizer is an annotator based upon machine-learning techniques applied to training documents (see text).
Relation Extractor	Identifies syntactic clauses containing noun and verb phrases.	Extracts clauses consisting of subjects, verbs, and objects. Takes shallow parsing annotations as input.

proteins) and facts about them (e.g., a gene expresses a protein, one protein activates another protein, a gene cluster appears implicated in a disease syndrome, etc.). Table 1 summarizes features and functions of the annotators that are used in the "Text Analysis Engine" component shown in Figure 4.

In BioTeKS, text annotators have a standard implementation specified by UIMA, in which annotators create, use, and interoperate via the Common Analysis System (CAS). CAS annotations are structures consisting of attribute-value pairs, where the attribute is some type of classifying or identifying description (e.g., a linguistic category like "noun," or a semantic category like "gene") and the value is a span of text in the document to which this annotation applies. Each annotation can have additional properties associated with it as well, such as the location of the span of text in the document. Each annotator takes a CAS structure as input and returns a modified CAS structure as output. Text annotators are executed in the text analysis engine shown in Figure 4, and multiple annotators can be organized in a pipeline.

The current set of BioTeKS text annotators are shown in Table 1. These annotators focus on information extraction; that is, identifying the location, category, and properties of the names of significant biomedical entities and identifying certain relations between these entities. However, to analyze entities and relations requires exploiting more generic natural language processing (NLP) analyses, beginning with tokenization and part-of-speech tagging of text strings. These are discussed in the next section.

UIMA also specifies various services and methods for collection-processing management, resource management for linguistic and other resources needed by annotators (e.g., dictionary or ontology resources), and CAS consumer methods for translating CAS annotations to other forms of data that can be indexed and stored (see Figure 4). Neither UIMA nor BioTeKS have functions for accessing document collections in the sense of crawling. (Figure 4 shows document access and management as external to the text-analysis component). However, BioTeKS does adopt the UIMA collection-processing scheme for implementing "collection reader" functions, for feeding documents from a compiled collection into a text-analysis engine. A collection reader parses each input document and initializes a new CAS structure containing the initial flat text on which annotators will operate, as well as optional document meta-data (e.g., title, author, etc.). BioTeKS has a collection reader specialized for MEDLINE abstracts, and a reader for aspects of patent documents. In both cases, the documents are initially available as XML documents with labeled fields for document-level meta-data, such as, "title," "author," and "date," as well as fields for extended segments of text containing the contents of the documents (e.g., MEDLINE abstract, patent abstract or claims, etc.).

NLP annotators. BioTeKS includes the following generic NLP text annotation methods:

- LanguageWare* linguistic engine
- Part-of-speech (POS) tagger
- Finite state transducer (FST) with shallow parsing syntax rules

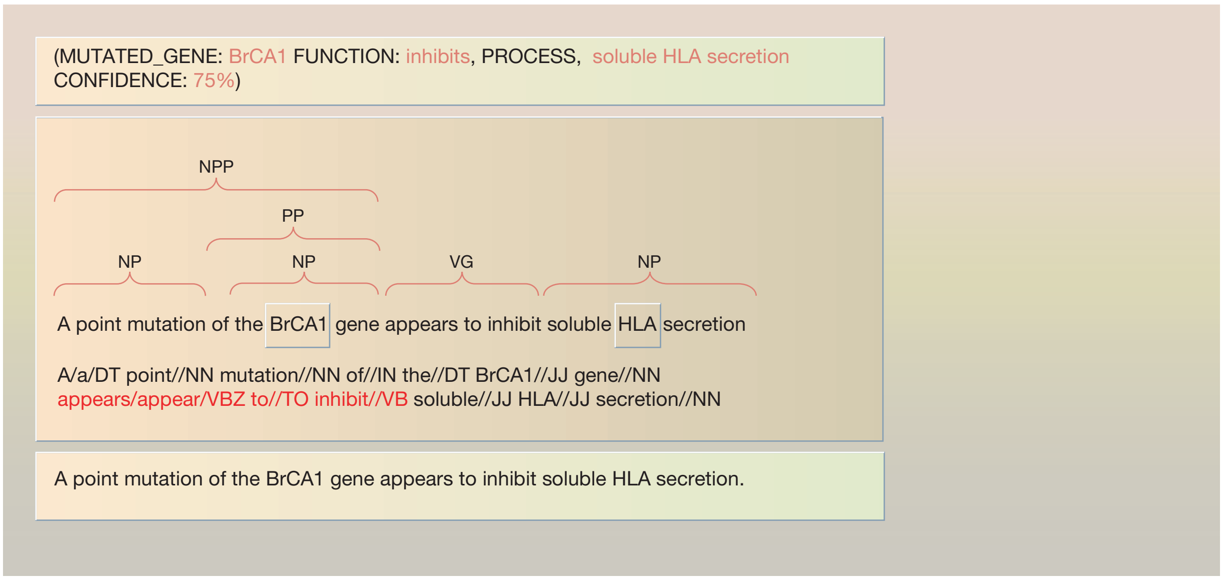
The LanguageWare linguistic engine[24] segments text into tokens and sentences, using a specific text annotation model. The LanguageWare tokenizer combines dictionary lookup with algorithmic processing to segment input text into distinct lexical units.[25] LanguageWare dictionaries also contain additional lexical information that can be associated with the lexical items identified as part of segmentation, such as a word's lemma or part of speech. This lexical information is useful for the subsequent annotation processes, including disambiguation of POS tags.

The POS tagger and FST component annotators are research-enhanced annotators based on an earlier text analysis engine developed by IBM Research, called *Textract*, which was available in the IBM Intelligent Miner* for Text (IM4T) software product.[26] (Textract components are described in more detail in Reference 27.) These two annotators, along with the LanguageWare tokenizer, use a common "text annotation framework" (TAF), also described in Reference 27. TAF specifies a set of CAS annotation types appropriate for multiple levels of linguistic processing, for example, tokens (strings), terms (including multiword phrases), sentences, clauses, and so forth, and properties of these linguistic objects, such as the character location of the span of annotated text in a document, part of speech for the span, and so forth. Most BioTeKS annotators interoperate through this annotation type system, or through annotation types derived from this set.

The output of the tokenizer consists of CAS annotations which form the input to the POS tagger. The POS tagger uses a statistical language model, created during a separate training phase, to disambiguate the possible POS tags based on the language model. Statistical methods are also used to determine a POS tag for tokens which are not in the language model. The tagger needs to be trained manually to develop this statistical language model by using a corpus annotated with correct POS tags.[28]

The shallow parser analyzes syntactic relationships among these POS-tagged terms, and produces a shallow syntactic parse, which groups tagged terms into noun, verb, prepositional phrases, and so forth, and in some cases, identifying the "subject," "verb," and "object" clause structure of sentences. Figure 5 shows an example of shallow-parsing results for a sentence, indicating POS tags and syntactic groups of tagged terms (NP is noun phrase, VP is verb phrase, PP is prepositional phrase, NPP is noun phrase with prepositional phrase, and VG is verb group). BioTeKS does not generate the ideal semantic representation in the top frame, but it does generate the interme-

Figure 5 Shallow parsing results for a sentance in PubMed abstract 12568865


(MUTATED_GENE: BrCA1 FUNCTION: inhibits, PROCESS, soluble HLA secretion CONFIDENCE: 75%)
NPP
PP
NP
NP
VG
NP
A point mutation of the BrCA1 gene appears to inhibit soluble HLA secretion
A/a/DT point//NN mutation//NN of//IN the//DT BrCA1//JJ gene//NN appears/appear/VBZ to//TO inhibit//VB soluble//JJ HLA//JJ secretion//NN
A point mutation of the BrCA1 gene appears to inhibit soluble HLA secretion.

diate shallow parse. This syntactic parse is useful for certain types of relation extraction because verbs typically describe relations between named entities in a sentence.

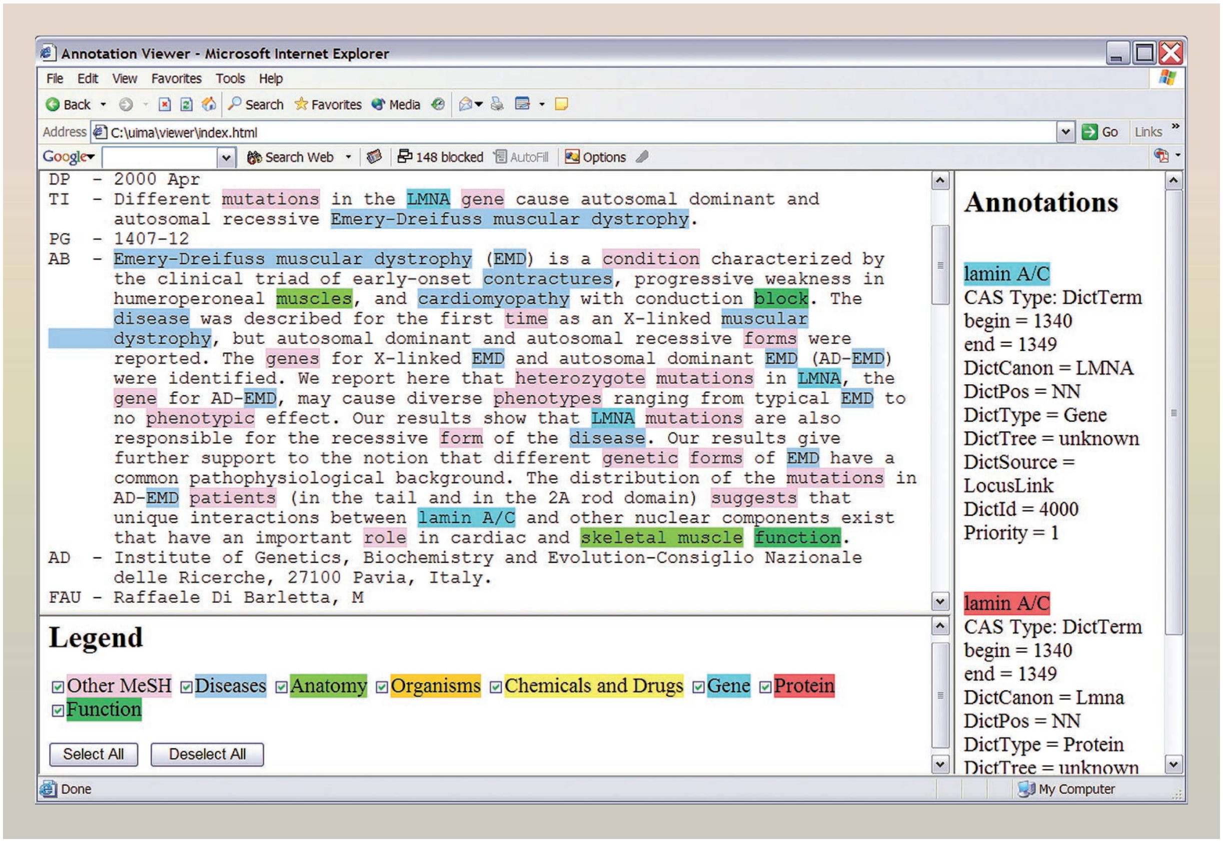
The shallow parser is realized as a cascade of finite state grammars running within a generalized FST engine. This transduces one sequence of symbols into another.[8, 22] For example, rules corresponding to English grammar are used by the FST to compose a sequence of word tokens (with parts of speech, such as noun or adjective, disambiguated by a statistical tagger) into noun phrases (NPs) to combine NPs and other phrasal units into syntactic arguments, and to assign grammatical roles (such as "subject" or "object") to these arguments. FST rules are declarative and capture linguistic regularities in human-readable form. An FST compiler compiles these rules into a runtime "state transition graph" executable that can be run efficiently against sequences of text fragments (e.g., document content) or, more abstractly, applied to sequences of annotations that may be of lexical, syntactic, or indeed any annotation type. In addition to its use for shallow parsing, the FST engine provides powerful general-purpose capabilities for manipulating text structures, syntactic or otherwise.

These annotators produce a rich set of generic linguistic annotations that capture key linguistic information which can be used by other BioTeKS annotators and applications, as we describe in the following. This information is useful outside the context of biomedical text. Indeed, pre-UIMA versions of the TALENT (Text Analysis and Language Engineering Technology) tools on which the TAF components are based have been used in applications involving customer call centers and other information.[13,29]

These annotators can be modified to work effectively on biomedical text. For example, standard POS taggers and shallow-parser rules are developed for sentence structures in generic text (e.g., news articles), and need to be modified for sentences and word-to-word statistical patterns characteristic of narrated physician reports or other kinds of biomedical text. In addition we should note that additional NLP annotators exist, including deep syntactic parsers.[30] However, we have not yet exploited these for life-science tasks.

Annotators for biomedical entity extraction. Entity extractors are annotators that identify the location of an entity name in a text and categorize the name relative to one or more knowledge resources like MeSH and UMLS** (Unified Medical Language System), both developed by the National Library of Medicine.[31] Examples shown in Table 1 include en-

Figure 6 MEDLINE abstract annotated with automatically extracted MeSH and gene names


Annotation Viewer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\uima\viewer\index.html
Go Links
Google Search Web 148 blocked AutoFill Options
Select All
Deselect All
Done
My Computer

```
DP  - 2000 Apr
TI  - Different mutations in the LMNA gene cause autosomal dominant and
      autosomal recessive Emery-Dreifuss muscular dystrophy.
PG  - 1407-12
AB  - Emery-Dreifuss muscular dystrophy (EMD) is a condition characterized by
      the clinical triad of early-onset contractures, progressive weakness in
      humeroperoneal muscles, and cardiomyopathy with conduction block. The
      disease was described for the first time as an X-linked muscular
      dystrophy, but autosomal dominant and autosomal recessive forms were
      reported. The genes for X-linked EMD and autosomal dominant EMD (AD-EMD)
      were identified. We report here that heterozygote mutations in LMNA, the
      gene for AD-EMD, may cause diverse phenotypes ranging from typical EMD to
      no phenotypic effect. Our results show that LMNA mutations are also
      responsible for the recessive form of the disease. Our results give
      further support to the notion that different genetic forms of EMD have a
      common pathophysiological background. The distribution of the mutations in
      AD-EMD patients (in the tail and in the 2A rod domain) suggests that
      unique interactions between lamin A/C and other nuclear components exist
      that have an important role in cardiac and skeletal muscle function.
AD  - Institute of Genetics, Biochemistry and Evolution-Consiglio Nazionale
      delle Ricerche, 27100 Pavia, Italy.
FAU - Raffaele Di Barletta, M
```

Legend

☑ Other MeSH ☑ Diseases ☑ Anatomy ☑ Organisms ☑ Chemicals and Drugs ☑ Gene ☑ Protein
☑ Function

Annotations

lamin A/C
CAS Type: DictTerm
begin = 1340
end = 1349
DictCanon = LMNA
DictPos = NN
DictType = Gene
DictTree = unknown
DictSource = LocusLink
DictId = 4000
Priority = 1

lamin A/C
CAS Type: DictTerm
begin = 1340
end = 1349
DictCanon = Lmna
DictPos = NN
DictType = Protein
DictTree = unknown

tity extractors for identifying genes, MeSH terms (also used for manual annotation of MEDLINE abstracts), drug names, and chemical-compound names.

Figure 6 shows a debugging tool used in BioTeKS to annotate in color specific categories of entities in a MEDLINE abstract. The legend at the bottom identifies semantic categories of words (e.g., "genes," "diseases," "chemicals and drugs"), and the data structure in the right frame is a CAS annotation for the selected term in the document. The annotation frame shows the CAS annotation for lamin A/C. Note that this is a variant of the gene LMNA, which appears in the title line (TI). Note also that the gene string is annotated relative to LocusLink,[32] a publicly available gene description database.

Entity extraction is a broad domain (see Reference 8), and there are multiple techniques available. BioTeKS is exploring three approaches to entity extraction, and Table 1 identifies for each annotator the general technique used to implement each annotator, namely:

- Pattern matching of terms (roughly string lookup) using a dictionary or database of known terms in some target category of terms (e.g., "MeSH" terms, LocusLink-derived "gene" names, etc.)
- Rules defined over a set of features or annotations characteristic of a category of terms
- Machine learning, based on human-created training documents containing correct examples of some target category of terms and also based on features or annotations associated with the target category of terms

The strategy in BioTeKS is not to build annotators for every possible biomedical entity. This is an open-ended task that typically requires access to specialized text sources (e.g., medical records) and domain expertise. Rather, we have developed examples of general techniques for a set of representative entities typical of bioinformatics (e.g., genes and proteins), medical informatics (e.g., drugs and disease indicators), and patent mining (e.g., drug and chemical names, disease indicators). New annotators can be built by extending this set of annotators by adding new dictionaries, FST rules, or training documents in the case of machine learning. This process is best done with domain experts, who can provide more in-depth expertise necessary for evaluating the quality of entity identification and for inferring how to iteratively improve the quality of annotators by adding terms and synonyms to dictionaries, improving rules, or developing more accurate training documents.

The Dictionary Lookup annotator uses a "dictionary" of entity names, each categorized in relation to a knowledge resource such as MeSH. For example, in addition to a dictionary of MeSH terms, we developed a dictionary for gene names, compiled from the public LocusLink database.[32] These dictionaries are stored as XML files that include the canonical form of biomedical entity names, as well as lexical variants, and other information, such as references to the source (database, authority, or "ontology") and an identifier of the entity name in that source.

The annotator applies pattern matching to match tokens in the text (potential names) to each item in the dictionary. The matching process is more than simple lookup because it can also handle variations in case and morphology (e.g., "Trk A" or "Trk-A"), including stemming (e.g., the common "stem" underlying plural vs. singular forms of a word).

The Dictionary Lookup annotator is actually several annotators, each specialized for specific dictionaries. Dictionaries can be built from MeSH and UMLS resources available from the National Library of Medicine[33] (NLM), as well as publicly available databases specialized for specific biomedical entities like proteins (Swiss-Prot[21]) or genes (LocusLink[32]). Note that many pharmaceutical and biotechnology companies have also developed internal and proprietary dictionaries of terms, including variants and synonyms. In general, dictionary tools need to be able to incorporate new dictionary resources, and the Dictionary Lookup tool can do so, using other dictionaries when they are properly formatted.

The BioAnnotator (see Table 1) categorizes noun phrases provided by a shallow parser as biomedical phrases when they match terms in UMLS, either as complete matches or as partial (substring) matches. BioAnnotator uses the LanguageWare linguistic engine described earlier to identify UMLS terms. This is done by replacing the default English language dictionary in the engine with a dictionary based on UMLS. BioAnnotator also has a rule-based component to identify biological terms not present in UMLS and to resolve certain types of ambiguity in extracted gene names (e.g., some gene names are also used to name proteins or nonbiological entities, such as "BIKE").[34] Dictionary Lookup and BioAnnotator are annotator options that have overlapping functions, but also explore different entity identification methods.

The Term Categorizer annotator elaborates on the semantic context of identified terms. For example, MeSH terms are categorized in a hierarchical taxonomy. This annotator optionally associates identified terms with additional information such as synonym and cross-reference information. Upstream applications can use this information in various ways, for example, to create a navigation function for browsing and selecting terms. This function could be bundled with Dictionary Lookup, but because it is an optional level of annotation, it is appropriate to keep it separate and invoked only when needed.

Entity extraction using dictionary lookup works well when domain experts can enumerate the names of entities of interest (as well as variants of these names). However, this is not always possible. The second approach to entity extraction is based on rules, not enumeration, where the rules can be developed by domain experts, either directly or by using machine-learning methods. The context in which a name occurs can sometimes provide feature cues for the semantic type of name for a term. Where this is the case, it is sometimes possible to write rules based on these features. The ChemFrag and DrugDosage annotators are examples of annotators based on rules.

The ChemFrag annotator combines regular expression rules that recognize organic chemical names with rules that assemble these fragments into larger descriptions. A small dictionary of prefixes and suffixes is used in some of the rules. An example of a recognized chemical fragment is *Pivaloyloxymethyl 1-ethyl-1,4-dihydro-4-oxo-7-(4-pyridyl)-3-quinolinecarboxylate*. Note that identification in this case only means categorizing chemical names as "chemical names," and does not mean identifying the specific chemical name in a standard resource (e.g., Reference 35). Rules are formal expressions, such as, "A fragment contains balanced parentheses or brackets and possibly, numbers and hyphens." ChemFrag is a hybrid consisting mainly of rule classification based on features, augmented with a small dictionary of known prefixes and suffixes.

The DrugDosage annotator (see Table 1) is also a hybrid annotator.[36] In DrugDosage, dictionaries are used to identify known drug names (e.g., "Ibuprofin") and strings associated with quantities and dosages (e.g., the quantity "20," and the abbreviation "mg" for milligrams). Rules classify co-occurrences of drug names, quantities and dosage abbreviations (e.g., "ibuprofen, 20 mg") as "drug and dosage" concepts. The rules for identifying these concepts use a modified version of the FST engine used in the TAF shallow parser. However, instead of compiling and using English language syntax rules, the rules for drugs and dosages are modified to apply to noun phrases that contain patterns of drug names, quantities, and dosage modifiers. Writing rules manually requires expertise and iterative refinement to achieve satisfactory levels of accuracy.

The BioTeKS team is also exploring machine learning (ML) approaches to entity extraction. ML approaches are especially useful when neither dictionaries nor explicit rules are easy or possible to build. ML approaches consist of a training phase involving the creation by humans of a training corpus, providing true examples of a category of entity to be learned (e.g., drug or gene names). The ML process automatically builds a classification model of the target category of term based on features associated with terms in the document context. These may be features of the term itself (e.g., distinctive characters, substrings, prefixes, or suffixes) or features of the linguistic and semantic context in which entity names occur.

There are several ML tools, some of which are available in the public domain, for example, the WEKA tools.[36] In BioTeKS, we are using an IBM Research prototype for interactive and semi-supervised learning.[37] The prototype uses a statistical-machine-learning algorithm called Generalized Winnow (an improved version of the standard Winnow algorithm), described in References 38 and 39. Some of the advantages of this tool are its interactive visual user interface for guiding the process of identifying true instances of an entity category based on confidence levels (estimates of in-class probabilities), the optional use of UIMA text annotators by the learning algorithm, and the use of the resulting classification model as a UIMA annotator for entity extraction.

Annotators for biomedical relation extraction. BioTeKS also provides the capability to extract what is expressed or predicated about specific entities, once they have been identified. Extracting predications means identifying relationships expressed between two or more terms describing simpler entities. The following are examples of biomedical relationships (where the bracketed and capitalized terms refer to semantic categories of entities related to each other):

1. [PROTEIN:] "ABP1" co-occurs with [PROTEIN:] "SRV2." (NOTE: the actual source sentence is "SH3 domain of ABP1p binds specifically in vitro to the proline rich segment of SRV2p.")
2. [RELATION:] located in, [SUBJECT:] collagen molecules, [OBJECT:] type XIII plasma membranes (of these cells)
3. [RELATION:] localized in, [SUBJECT:] [PROTEIN:] "PRP20", [OBJECT:] [CELL STRUCTURE] the nucleus
4. [MUTATED_GENE:] BrCA1, [FUNCTION:] inhibits, [PROCESS:] soluble HLA secretion (The original sentence is: "Apocytochrome c blocks caspase-9 activation and BAX induced apoptosis.")

As with entity identification, there are multiple approaches for extracting information about relations between entities, and BioTeKS is exploring at least three methods.

- Computation of mutual co-occurrence of terms ("unnamed relations" for short)
- Extraction of syntactic clauses using shallow and deep linguistic parsing, with additional filtering of relations based on the category of entities in the relations ("named relations" for short)
- Graph mining, based on compiling relationships with semantic and syntactic information, and searching for common structural patterns in subgraphs.

The first technique computes the frequency of co-occurrence of entity names within a window of text, for example, a sentence, paragraph, abstract, or other segment. Co-occurrences can be found in multiple documents, or multiple occurrences can be found within a single document. We call these "unnamed relations" because they identify a likely association between two or more terms, but cannot identify or name the specific relationship. An example is item 1 in the examples listed previously (a co-occurrence relation). Unnamed relations can be expressed in a number of ways, including in a database triplet consisting of the associated entities and a quantitative measure of the strength of association (see Reference 40 for details).

The next two analysis methods use linguistic information to identify relations, including information about verbs connecting two entities. The name of the verb can be used to name the relationship, and hence we call these "named relations." The Relation Extractor annotator (see Table 1) extracts syntactic relations, using the shallow-parsing results of the Shallow-Parser annotator to extract syntactic clauses connecting nouns and verbs. In this case, the relation is expressed as a verb connecting a subject noun and an object noun, as shown in examples 2 and 3. These examples are essentially a compilation of verb, subject, and object noun phrases extracted from the Relation Extractor annotator.

Once a set of such relations is extracted, it needs to be further analyzed to focus on specific relations of interest. For example, example 4 is an instance of a set of relations filtered by identifying verbs that are characteristic of gene functions. Identifying gene functions is a key bioinformatics task, and the bioinformatics research community has provided a mechanism for manual creation of descriptions of gene functions, including a reference to a specific MEDLINE record where this function is described in detail. These descriptions are called Gene RIF (reference into function) descriptions. Example 4 is expressed as a sentence and also by a somewhat more formal representation labeling the entities and functions expressed by verbs such as "blocks," "activation," and "induced." This representation is shown only for illustrative purposes because there are currently no formal standards for creating Gene RIF descriptions (although Gene Ontology[41] would be a candidate). The National Institute of Standards and Technology (NIST) sponsored the first Genomics TREC (Text Retrieval Conference) track in 2003, with a focus on finding MEDLINE abstracts and Gene RIF descriptions for target genes using text information extraction and search methods. Our work on identifying gene functions is based on IBM's participation in this NIST-sponsored Genomics track.[42]

The Relation Extractor annotator is also being extended to include graph-mining techniques. Graph mining applies data-mining techniques to graphs that are derived from syntactic parse trees, where the graphs' nodes and edges represent terms and syntactic relations, respectively. Inokuchi has developed graph-mining algorithms to find salient and frequent patterns or subgraphs corresponding to interesting relations.[43]

Performance of BioTeKS annotators. It is reasonable to ask is how fast the BioTeKS annotators are in processing documents. In the case of MEDLINE abstracts, the corpus we are indexing currently consists of about 11 million abstracts, corresponding to 35 gigabytes of text. The performance depends on what annotators are applied in a text-analysis engine (see Figure 4), what is done with the annotations (e.g., are they used to index a search engine or load a database), the computing resources applied, and potential optimization methods. We are using a single XSeries* 335 Intel Xeon** 2.8 GHz processor with 1.5 GB of memory running Linux** release 9.0. Some idea of the throughput can be gained from the results of two analysis runs:

- In the first run, it took 40 hours to annotate the full MEDLINE corpus, using selected annotators and annotator translation applied to each MEDLINE abstract. Each MEDLINE abstract in XML format was parsed, terms and sentences were tokenized, and Dictionary Lookup was used three times to identify MeSH terms, expanded gene names, and gene function verbs. CAS consumers were used to translate annotations to an indexable format. An additional 120 hours were needed to create a Juru XML index (see "Semantic text search") from the MEDLINE abstract content, meta-data, and extracted annotations.
- In the second run, 360 hours were needed to annotate the full MEDLINE corpus, using selected annotators and annotation translation. Each MEDLINE abstract in XML format was parsed, terms and sentences were tokenized, and POS tagging, shallow parsing, and Dictionary Lookup for MeSH terms were applied. CAS consumers were used to create indexing input for text search (without actually creating a Juru XML index), and for creating noun-phrase feature files for on-demand clustering of document search results.

The difference between these two cases is the use of POS tagging and shallow linguistic parsing. It is generally the case that such processes are computationally intensive compared to other processes: for example, the current shallow parser can parse 14 documents per second, as compared to Dictionary Lookup, which can process between 200 and 1558 documents per second, depending on the size and complexity of the dictionary entries. However, indexing the entire MEDLINE corpus (or any other corpus) is likely to be done very infrequently, and once it is done, the indexes can be incrementally updated as new documents are added to the source collection. There are numerous parameters influencing processing throughput, and several optimization strategies are being explored.

Quality of BioTeKS entity and relation annotators. It is important to evaluate the quality of entity and relation extraction, but it is also quite difficult, given the state of the art in these technologies. Evaluation requires the manual definition of a test bed of accurately categorized entities or relations, against which the results of text annotations can be compared. This is difficult because of the inherent difficulty of unambiguously assigning meaning to language expressions (humans do not always agree on how to categorize a term or phrase), and because there are virtually no comprehensive test beds against which to compare performance (see Reference 10 for a discussion of this state of affairs). Nonetheless, the BioTeKS project is pursuing the evaluation of selected annotators. The following are some examples.

The BioAnnotator tool described earlier identifies "biomedical concepts" corresponding to noun phrases that contain one or more keywords in one or more UMLS thesauruses (including MeSH keywords). An evaluation against the GENIA test bed of 670 MEDLINE abstracts[44] produced standard precision, recall, and F-values[45] of 0.87, 0.94 and 0.90, respectively, for approximate matching (i.e., finding phrases that have any GENIA term in them; for a full report, see Reference 34). These are reasonable figures, although identifying more specific categories of terms, such as specific MeSH or UMLS terms, may be more difficult and is the subject of ongoing investigation.

An unpublished evaluation of the ChemFrag annotator evaluated the accuracy of identification of organic chemical names in a manually annotated test bed of ten patent documents. The evaluation produced precision, recall, and F-values of 0.91, 0.94 and 0.92, respectively. These are also reasonable figures although the sample is small and there is considerable room for improvement. Note that identification in this case means only categorizing names as any "organic chemical names," and does not mean identifying the structure of that compound (see Reference 35).

Regardless of these results, information extraction (entity names and relations) is very difficult, and a very active and challenging domain of research. A few examples can indicate the difficulties. Annotators need to identify lexical variants of an entity name based on case and stemming. A case variant is exemplified by "trk A" vs. "Trk A." Handling stemming variants means identifying the common stems expressed in variations resulting from factors like pluralization, as in "actin filament(s)," or by tense variation as in "bind(s/ing)." The entity annotators described earlier include an approach for handling case and stemming variants, but there are also ambiguities that require taking the larger context into account. In a biomedical context, case may carry information; for example, the term "BIKE" capitalized is likely to be a gene name, whereas "bike" or "Bike" is likely to be the vehicle. Annotators need to be able to take into account the context in which ambiguous terms occur in order to resolve the ambiguity.

Annotators need to be able to handle partitive phrase variants, that is, terms that are phrases, and that may match part of larger phrases of biomedical interest. A partitive phrase variant is exemplified by finding the term "dendritic" in the phrase "dendritic structures" in a MEDLINE abstract, but not finding this specific phrase in MeSH, while many other phrases with the term "dendritic" (e.g., "dendritic cells") do occur in MeSH. The entity annotators described earlier can also handle partitive phrase variants, but again, there are policy issues to resolve. In some cases, we may want only strict matching of phrases relative to some resource like MeSH, whereas in other cases, we may want to be made aware of phrases that contain biomedically relevant terms (e.g., "dendritic structures") even if these specific phrases do not actually appear in MeSH. The BioAnnotator uses this criterion to identify "biologically interesting" phrases.

Annotators need to be able to resolve other types of word-sense ambiguity. Names of entities can be ambiguous without taking context into account. This is especially true of gene names, which often do not conform to simple naming conventions, and which can be ambiguous with respect to other common English language terms. A simple dictionary lookup finds the gene names "FHC" or "OF" in the following sentences. Both gene names are represented in a gene dictionary based on LocusLink:[32]

- "Recent genotype-phenotype correlation studies in familial hypertrophic cardiomyopathy (FHC) have revealed that some mutations in the beta-mysoin heavy chain (BMHC) gene may be associated with a high incidence of sudden death and a poor prognosis." (from PubMeD** PMID 8655135)
- "BACKGROUND OF THE INVENTION" (from a section title of a US patent)

In the first sentence, "FHC is also an acronym for "familial hypertrophic cardiomyopathy," and in this context "FHC" is not a gene name. The "OF" gene in the second example is actually a preposition in a patent document title, which is capitalized. These are examples of word-sense ambiguity, where the same term can mean different things depending on the context in which it occurs. This is a well-known and classic problem in NLP.[8] Identifying the names of entities often requires additional word-sense disambiguation processes, which typically entail analyzing the context in which names occur. The Bio-Annotator annotator includes methods that use certain types of features to select the correct word sense. An example is exemplified in the PubMeD sentence above: ". . . in the beta-mysoin heavy chain (BMHC) gene. . ." The presence of the term "gene" in the phrase ". . . (BMHC) gene" is additional evidence that "BMHC" is indeed the name of a gene.

BioTeKS includes annotators that compare the results of multiple annotators and choose the most plausible annotation. For example, the POS tagging and shallow-parsing annotators assign the syntactic annotation "PREPOSITION" to the "OF" keyword in the patent example above. A disambiguation rule that requires gene name annotations to apply only to keywords that are annotated with the syntactic annotation "NOUN" will reject the "GENE" entity name annotation for this string annotated as a "PREPOSITION." Similarly for the "FHC" gene, an abbreviation annotator can associate phrases like "familial hypertrophic cardiomyopathy" with acronyms like "FHC" in close textual proximity.[46] A disambiguation rule gives precedence to the "ABBREVIATION" annotation, compared to the "GENE" entity name. Disambiguation rules must be written for a variety of cases, and we are exploring these in the BioTeKS project.

Like entity extraction, evaluating the accuracy of extracted relations is quite difficult. It is necessary to develop test beds with real examples of relations against which to compare relation extractors. The BioTeKS project has data for two evaluations of relation extraction. One evaluation focused on protein-protein interactions in a manually annotated set of 573 MEDLINE abstracts based on a protein interaction database developed by the Munich Information Center for Protein Sequences.[47] We conducted an analysis of mutual co-occurrence, and filtered the unnamed relations by selecting relations with "protein" entity names as subject and object nouns. This resulted in precision and recall measures of 0.30 and 0.40 respectively.[48] An analysis of the errors points to the complexity of identifying protein names and limitations of the MEDLINE abstracts, as opposed to full source literature. For example, we identified problems relating to handling complicated synonyms, such as the synonyms for "SRV2," which are "Srv2p," "SRV2p," "CAP," "CAP1," and so forth. We developed functions to identify protein complexes, such as those represented as "prot1-prot2," which also express protein interactions.

Other problems are related to complex syntax, as in a sentence like "Proteins A, B and C interact with D." In this case, mutual co-occurrence might find all pair-wise relations among A, B, C and D, but only those connecting A, B and C with D may be valid. We discovered that many of the relations we detected which had not been tabulated actually were discussed in the abstracts, but had not been tabulated by the Munich Information Center for Protein Sequences database because they were not reports of new experiments. Finally we eliminated missing relations, which were not discussed in the abstracts but only in the full papers. After taking all of these factors into account, we found a more respectable precision of 0.91 and recall of 0.84.

A second evaluation involved finding MEDLINE documents (and sentences in them) that describe gene functions for a target gene. We alluded to this task in our earlier discussion of the NIST Genomics track.[42] The track organizers created a test bed of 500 000 MEDLINE abstracts that were associated with manually created Gene RIFs describing the functions associated with genes described in the abstracts. Participants in the Genomics track had two tasks. The first task was to find relevant abstracts and Gene RIFs, given a target gene, where a relevant abstract is one that describes a function of the gene and, in particular, is cited in a Gene RIF entry for that gene. For

Table 2 Examples of CAS conversions in BioTeKS

1. *CAS to index specification*: Generates an indexing input for a text search engine. The text search engine used in BioTeKS indexes conventional content keywords indexed in the full-text index, sentence-boundary annotations, and a specifiable set of "semantic" annotations extracted by the text annotators.
2. *CAS to database index*: Generates a database load file for batch database indexing of text annotations. The database schema indexes document meta-data (e.g., title, author, date, etc.), extracted terms and statistics (e.g., location in source document), and semantic identifying information for terms (e.g., MeSH identification code).
3. *CAS to noun phrases*: Generates a formatted flat-file representation of indexing features needed as input for application-level clustering of document collections, such as that resulting from a text or database search. These cluster features could also be stored in a database.
4. *CAS to XML file*: Generates XML for the document text, with text annotations represented as inline XML-tagged strings and used for viewing entity annotations (e.g., MeSH terms) in a Web browser, with annotated terms highlighted and linked to their CAS annotation data structures. (see Figure 6)

the 50 genes provided as test queries by NIST, the BioTeKS team achieved a mean average precision of 0.28. The second task was to automatically extract the best sentence describing a gene's function from an article known to describe the gene's function. Gene RIF descriptions (created by biology experts) were used as a "gold standard" for describing gene functions. The BioTeKS team achieved .505 on a measure of similarity computed between the text summaries generated by the BioTeKS system for each MEDLINE abstract (a sentence selected from the abstract), and the "standard" description contained in the Gene RIF text.[42]

Relation extraction based on co-occurrence statistics, syntax, or even graph mining is a good start but may not be abstract enough to represent the deeper semantics of what is being asserted or predicated about the entities named in the relations. Annotators need to map varied forms of relationship expression to a common underlying relationship. For example, a purely syntactic analysis would extract as separate relations the following active and passive forms of a relationship: "Trk A expression is inhibited by factor X" vs. "Factor X inhibits Trk A expression." To extract the common underlying relationship, a top-down approach may be necessary, using a more knowledge-based and ontology-based approach for representing relations. The analysis of semantic relations is an active area of research in the bioinformatics and computational linguistic research communities.[17,49,50]

Indexing and accessing text annotations. The text annotators described in the previous sections all produce CAS annotations. Typically, these annotations need to be stored for each document in the collection so that they can be accessed easily for further collection-level text-mining processing and applications. In UIMA, CAS annotations are converted to other forms of data by implementing CAS consumers. Examples of CAS conversions in BioTeKS are shown in Table 2, and include extractors which create formatted input for indexing a database and text-search engine, noun phrases for a clustering engine (described later), and an XML file of the annotations represented as XML tags embedded in the source document (and viewable as Web pages, as in Figure 6).

CAS consumers iterate, filter, and translate CAS annotations into other formatted data, including strings and flat files. The details depend on the text-analysis data needed by applications. For example, the database schema alluded to in the figure includes a table for storing extracted terms and their offsets (locations) in documents. This information is used to compute mutual co-occurrence statistics for text mining applications.[40] In other cases, storing text-analysis results is convenient for performance reasons. For example, the noun phrases used for document clustering can be extracted in real time for an ad hoc collection of documents (e.g., resulting from a search), but computing and storing them ahead of time in a specialized file format improves overall clustering performance.

Biomedical ontology resources. An ontology is a conceptual framework for defining the basic classes of entities in some domain of knowledge, the relationships these entities have to each other, and the organization of concepts in terms of higher-level concepts, typically taxonomic in nature.[51,52] The term *ontology* is often used to refer to a range of linguistic and conceptual resources, from thesauri and dictionaries containing records defining key terms, synonyms, and so forth in some domain, to taxonomies that classify names and phrases in higher-level categories, and formal knowledge representations that might support automatic inferences and certain types of reasoning.

In biology, an ontology like MeSH[33] provides a set of broad-based, multidisciplinary concepts and categories that biomedical experts (librarians) use to annotate the content of literature in terms of key concepts describing genes, proteins, cell function, anatomical objects, diseases, and so on. The Gene Ontology[41] (GO) is a more specialized and semantically richer ontology that organizes knowledge about genes and cell functions associated with genes. Relations may be taxonomic (e.g., "Trk A" is a "receptor"), or part-whole (e.g., "cell membrane" is part of "cell").[41] Ontologies have to be built and maintained, and this is a difficult, labor-intensive, and expertise-intensive task. The National Library of Medicine supports major ontology development efforts (the UMLS metathesaurus[33,53]), as do many other research institutions (e.g., the GENIA Project[11]).

As discussed earlier, we use MeSH to create a lookup dictionary of terms expressing biomedical concepts. Dictionaries are not the only way that entity names can be identified (rules and machine-learning methods are among the alternatives), but they are an effective first approximation. Ontology information can also be stored in a relational database to support database queries on this information, and BioTeKS includes software components for loading MeSH and UMLS information into database schemas.

Application prototypes and application-enabling engines

The value of the annotations produced by BioTeKS is a function of the applications that use these annotations. In this section, we briefly highlight several application prototypes we have built with BioTeKS to indicate how it can be used. We have focused on two broad classes of applications: text searching and text mining. Searching refers to finding collections of documents that meet some search criteria. Text mining refers to analyzing collections of documents at a more fine-grained level, to identify, for example, relationships and trends among specific topics and concepts expressed within documents in a collection. Text mining can also be applied to the results of a search or to collections of documents derived from other kinds of processes. The following are several examples.

Semantic text search. One of the most basic application functions is searching for documents, for example MEDLINE abstracts, using biomedical terms. The focus in BioTeKS on generating rich semantic annotations of terminology in biomedical documents suggests the opportunity for text search to index not only keyword content, but also the semantic annotations associated with those keywords. That is, we want to explore the value of indexing and searching not only keywords corresponding to a specific gene name, such as "LMNA," but also indexing and searching annotations associated with these keywords, such as the category annotation "gene" associated with "LMNA." BioTeKS uses an IBM Research text search engine called Juru XML[54–56] to explore the indexing and search of annotations generated by BioTeKS annotators.

Semantic search can refer to many things,[8] but for our purposes it means simply indexing semantic information (in our case expressed as annotations) associated with keywords, and using this in the search process. Juru XML can index and search both keywords (based on text content) and semantic annotations of text keywords expressed as XML tags for keyword annotations, initially stored in CAS annotations.[54,56] The results of a Juru XML search are returned as a list of documents or document components, ordered by their relevance to the original query terms. For example, we can index all "sentences," the entity names contained within the span of these sentences, and other syntactic phrases (typically verb phrases) expressing biological functions of interest. This allows us to form queries on structures more closely approximating certain types of relations, such as protein-protein interactions.

For example, the following query, expressed in XML tags, will find sentences that contain keywords annotated as "proteins" and syntactic phrases that contain keywords annotated as "biological function" (typically verb phrases like "binds to" or "inhibits"):

```
<Sentence>

   <Protein>SRV2</Protein>
   <Function></Function>
   <Protein></Protein>

</Sentence>
```

In this query, the user wants to see MEDLINE abstracts that contain sentences with a specific protein "SRV2," any other protein, and terms and phrases that have been annotated by BioTeKS as "biological functions." The specific annotator used for annotating sentences in this way is the Dictionary Lookup annotator (see Table 1), using dictionaries of protein names and biological function terms. This query does not ensure that the abstracts which are found will contain an actual protein-protein interaction, but pending evaluation, we believe such queries will greatly increase the likelihood of finding this combination of semantically annotated terms, and hence find actual interactions of interest.

Document clustering. Document clustering is a way to organize document collections (such as those derived from search results) in topical clusters. Clustering can complement text search or any other function that compiles collections of documents as part of an analytic process. We previously discussed an example of document clustering in the context of the Bio-Dictionary tool[12] (shown in Figure 2). The role of BioTeKS, as we indicated, is to extract text features, such as noun phrases, for input to the clustering-engine algorithm. These noun phrases are extracted using the shallow linguistic parser, and this annotator in turn uses tokenization and POS tagging annotators.

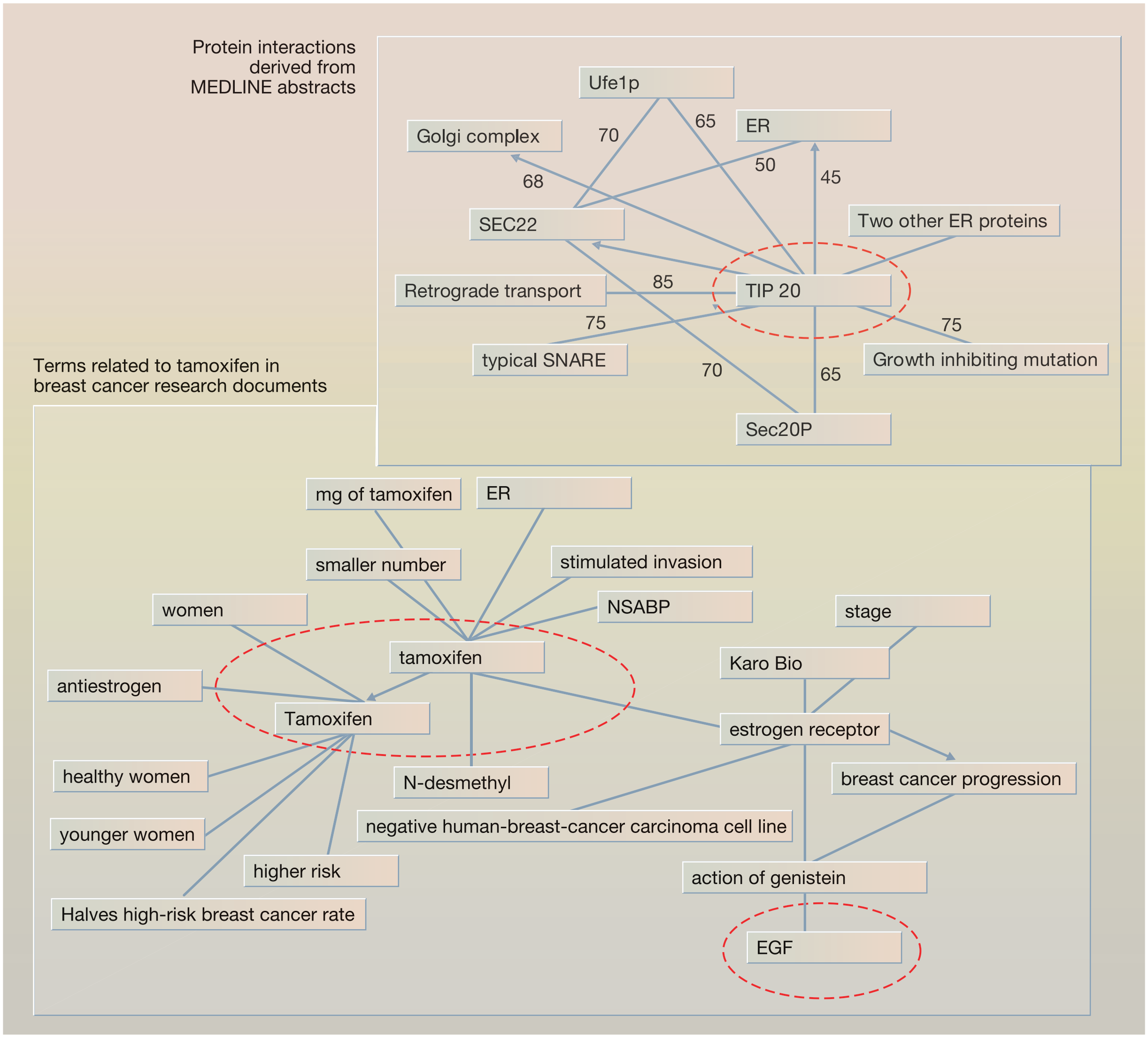
There are numerous clustering methods and tools, and we have experimented with several approaches in BioTeKS for different purposes. The clustering engine in Figure 2 uses a so-called subspace projection clustering method.[57] We also have used this clustering engine to organize the results of the text search engine described in the previous section. Other clustering methods available in BioTeKS include a model-based hierarchical clustering tool[58] and a cluster tool called "SearchEssence,"[59] which create a concept hierarchy from the collection of results returned by a search and label the clusters using the names of entities extracted by other annotators.

Association mining using MedTAKMI. MedTAKMI (Medical Text Analysis and Knowledge Management) is a version of the TAKMI* (Text Analysis and Knowledge Mining) system for mining relations and trends among different categories of named entities in a text collection based on search.[13] Users select a collection of documents (e.g., MEDLINE abstracts) and one or more categories or subcategories of entity in an ontology, such as MeSH, and then invoke an analysis of associations between these categories of entities. Association analysis can be presented to end users as a spreadsheet table. For example, a user might want to view associations between two selected term categories like "membrane proteins" and "organs and diseases." Cells in the table corresponding to terms that co-occur relatively more frequently than some background value are highlighted. Links can be provided from a cell to documents with the corresponding associations. Association mining and other MedTAKMI text-mining functions, including trend analysis, term distribution analysis, and so on, are described more fully in the paper by Uramoto et al. in this issue.[60]

Analyzing mutual co-occurrences among terms using lexical networks. Lexical networks are visualizations of the unnamed relations between extracted terms which we described earlier.[40] Nodes in a network graph represent the names of entities (e.g., proteins), and the links represent relatively strong statistical correlations between these entities within a sentence or paragraph (e.g., protein-protein interactions). Lexical networks allow pair-wise term relations computed in documents to be compiled into longer sequences that span multiple documents. The meaning of these longer sequences is not always clear and must be treated cautiously, but the links do provide clues to possible interesting connections. In some cases, these connections are genuine discoveries. For example, we replicated Swanson's discovery of a connection between "Raynaud's disease" and "fish oils" using lexical networks,[61–64] and we have explored the use of lexical networks in life-science demonstration prototypes. Figure 7 shows two examples of lexical networks. Term-to-term relations are generated from text-analysis data, which is extracted using BioTeKS text annotators. Numbers are strengths of associations. One network shows connections between a set of protein interactions extracted from a set of 600 MEDLINE documents. For example, "TIP20" is linked to "Sec20P" with an association strength of 65. The latter in turn is linked to "Sec22." Other biomedical phrases, such as "Golgi complex," also co-occur with these protein names in MEDLINE abstracts. The second network shows terms that co-occur with the drug name "tamoxifen" in the context of a collection of documents retrieved from a text search engine using the query "breast cancer." The connection between "tamoxifen" and "EGF" ("epidermal growth factor," circled in red) is of special interest.

In BioTeKS, mutual co-occurrence computations are based on text annotators that extract terms (e.g., protein names like "TIP20" and chemical names like "tamoxifen") and their location in documents. CAS consumers index these extracted terms for each MED-

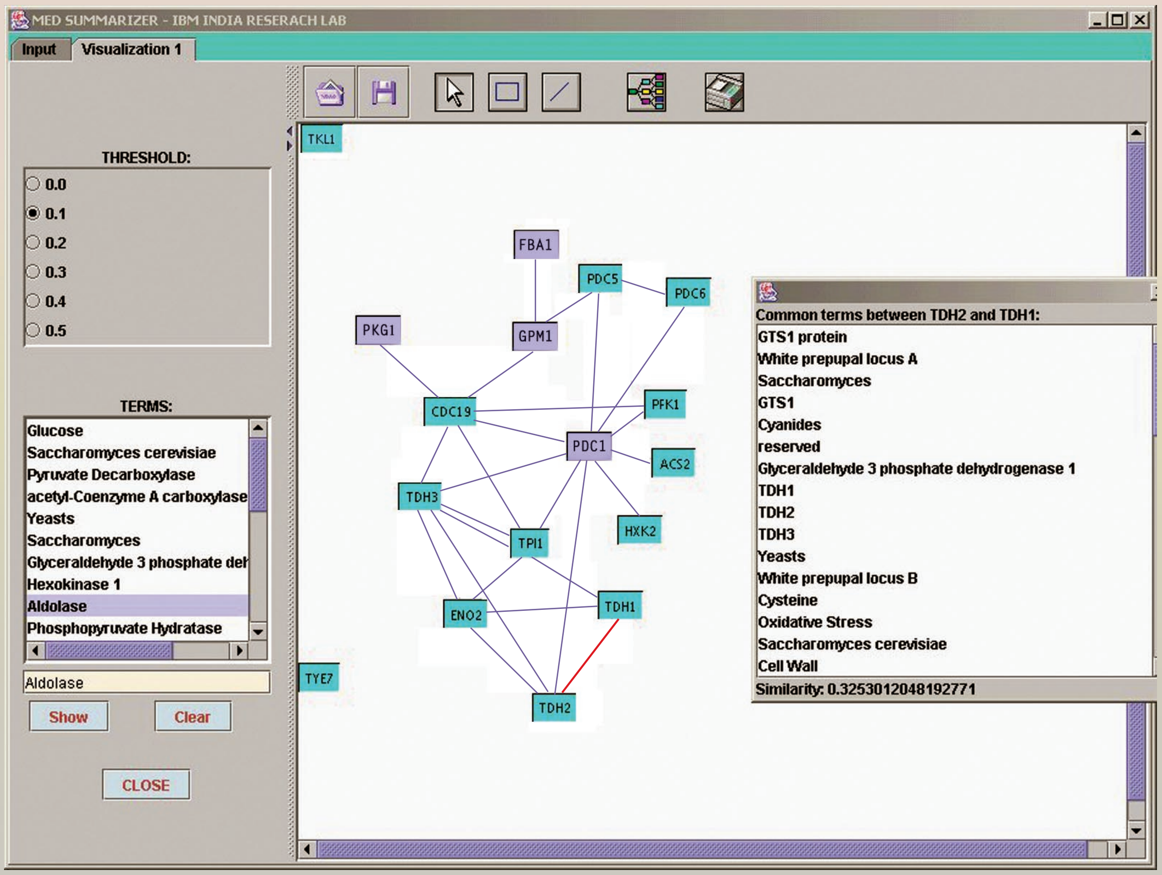
Figure 7   Lexical networks showing terms and relations derived from a collection of text documents



LINE document in a relational database. Mutual co-occurrence is a text-mining method applied to a collection of MEDLINE documents and implemented as a database query on the text data stored in the database for this collection.[40] These computations can be performed for predefined collections or subcollections of documents and on the unnamed relations also stored in a relational database for later access by an application.

Analyzing gene clustering using the MedMeSH Summarizer. MedMeSH is an application prototype that takes as input a cluster of gene names derived from a micro-array expression analysis, retrieves MEDLINE abstracts that contain those genes, and clusters MeSH terms extracted from those abstracts in ways intended to suggest the functions associated with these genes, at least insofar as these have been expressed in the literature as MeSH keywords. The idea is to organize MeSH keywords into topical clusters that can help to explain why these genes react collectively.[65] Figure 8 shows a sample screen shot of a MedMeSH Summarizer cluster analysis. The gene cluster is derived from gene expression data obtained from micro-array experiments. Keywords

Figure 8 Sample results of cluster analysis by MedMeSH Summarizer



THRESHOLD:
- ○ 0.0
- ● 0.1
- ○ 0.2
- ○ 0.3
- ○ 0.4
- ○ 0.5

TERMS:
Glucose
Saccharomyces cerevisiae
Pyruvate Decarboxylase
acetyl-Coenzyme A carboxylase
Yeasts
Saccharomyces
Glyceraldehyde 3 phosphate del
Hexokinase 1
Aldolase
Phosphopyruvate Hydratase

Aldolase

Common terms between TDH2 and TDH1:
GTS1 protein
White prepupal locus A
Saccharomyces
GTS1
Cyanides
reserved
Glyceraldehyde 3 phosphate dehydrogenase 1
TDH1
TDH2
TDH3
Yeasts
White prepupal locus B
Cysteine
Oxidative Stress
Saccharomyces cerevisiae
Cell Wall
Similarity: 0.3253012048192771

in the TERMS window are MeSH terms extracted and clustered from MEDLINE abstracts containing the gene names in the cluster. The figure shows that terms like "glucose," "yeast," and "hexokinase" are associated with the given set of genes. These terms indicate that these genes are involved in glycolysis, the process by which glucose is broken down in the cells of all higher animals.

The input genes are shown as the nodes of a graph. The input gene cluster for this example is a group of yeast genes encoding the enzymes involved in glycolysis. If the similarity between two genes is greater than the threshold (which can be specified by the end user), an edge is drawn between the nodes. Clicking on an edge shows the terms common to the genes, as expressed in MEDLINE abstracts that refer to the genes in the graph. For example, the terms common between TDH1 and TDH2 are shown in the popup window when you click on the link shown in red.

Building applications using BioTeKS. The application prototypes we have described all have a similar general design. Each application has one or more functions, such as document search, document clustering, or text mining of associations or statistical co-occurrences between terms across a collection of documents. Each function requires structured text data as input and computes new information based on it. The input text data is extracted using one or more text annotation methods executed by BioTeKS. Typically, text-mining methods apply to text data for a collection of documents, and therefore, text data must be accumulated across documents in a collection and stored in a persistent form. The role of BioTeKS is to apply a sequence of text annotators (orchestrated in the "Text Analysis Engine" in Figure 3) to documents in a collection, and to provide the translation methods (CAS consumers) to create text data formats (e.g., database load files) to index the text data in a form that allows it to be easily accessed by upstream text-search and text-mining applications, such as MedTAKMI or document clustering.

In this sense, BioTeKS is a middleware system that can be used to build applications and that can be used for many application-level tasks. Many important details of how BioTeKS is used will vary with different target application requirements. For example, different sets of annotators will be needed for different application-level functions. We also anticipate that text annotators built for BioTeKS will need to be customized or extended for specific research or potential customer applications. We know, for example, that pharmaceutical companies have internal knowledge resources including document collections and thesauri of terms and concepts proprietary to how these companies conduct research and development. Using BioTeKS to solve specific problems would require incorporating additional resources (dictionaries, etc.) into the BioTeKS annotators.

The value of BioTeKS in this applied setting is twofold. Firstly, it casts each application scenario into a similar sequence of application-design considerations—each application implies one or more types of documents (we have focused on MEDLINE abstracts), one or more types of text data, a configuration of one or more text annotators that can extract relevant types of text data, and one or more CAS consumers for translating annotations into persistent forms for convenient access by upstream applications. The database model implied for the text-mining applications discussed earlier may or may not be useful for other applications. Secondly, the value of BioTeKS is in the quality of its specific text annotation methods. The development and refinement of these methods is an ongoing process, involving both software engineering and basic research.

Conclusions

BioTeKS is a significant technical initiative within the IBM Research laboratories to integrate and customize a broad suite of text-analysis projects and technologies targeting problems in the domain of biomedical text analysis. The goal of the BioTeKS text-analysis methods is to convert initially unstructured text information into structured text data, commensurate with structured data derived from sources other than text (e.g., gene names in micro-array experiments, or drug, treatment, and disease references in clinical records). The domain of text-analysis problems that BioTeKS addresses is very broad, and many problems are the subject of basic research initiatives in industry and academic institutions.

Accordingly, both BioTeKS and UIMA are works in progress. Future work will focus on research issues and on solving real-world problems. Research will focus on improving the quality of BioTeKS annotators for both biomedical entities and the facts and relations associated with them. We need to better exploit emerging biomedical ontology resources in the process of IE. In addition, we believe machine learning techniques hold great promise for improving the quality of IE. Improving text-analysis quality also requires the development of, or access to, test beds or "gold standards" for correct identification of biomedical entities and relations. Lack of such test beds is a problem in the bioinformatics research community, as we noted earlier (see Reference 10), and it is equally problematic in analyzing clinical records and patents. Finally, a key requirement for making progress is the use of text analysis for the kinds of real-world drug-discovery and biomedical-discovery tasks surveyed in the introduction. The domain of potential IE problems is very large, and progress will require focus and feedback that we believe implies significant collaboration with domain experts (and problem solvers) in biomedical research and development organizations in both academia and industry.

In this context, BioTeKS has two key advantages that make it a useful platform for pursuing research in biomedical text analysis. First and foremost, BioTeKS is implemented using the UIMA framework. This means that BioTeKS can support serious experimentation in the development of text-analysis methods. Second, IBM Research has invested in core text-analysis technologies, including machine learning approaches. Combined with the UIMA framework, BioTeKS provides a broad-based platform for tackling all the key text-analysis problems, whose solutions are essential to progress in life-science research and development.

Acknowledgments

BioTeKS is in large part a systems integration effort that builds on technologies and expertise developed by many groups in IBM Research. All these groups generously provided their time and expertise to deliver code to the BioTeKS team, teach us how to use it, and in some cases modify their code for our needs. These groups include the UIMA / JEDII team in Hawthorne (David Ferrucci, Adam Lally, and Jerry Cwiklik), the TALENT text analysis team (Roy Byrd, Mary Neff, Rie Ando, Bran Boguraev, and Youngja Park), the machine learning team (David Johnson, Sylvie Levesque, Tong Zhang, Frank Oles, and Brian White), and research management at various levels for support and technical direction (Alan Marwick, Marshall Schor, Arthur Ciccolo, Ajay Royyuru, and Kohichi Takeda). Additional colleagues at other IBM Research locations were involved in various applications cited in this paper. These colleagues include Ravi Kothar, Pankaj Kankar, Biplav Srivastava, Vishal Batra, Krishna Kummamuru, and Pasumarti Kamesam at India Research; Jeff Kreulen, Dan Gruhl, and Shivakumar Vaithaynathan at Almaden Research; and Tetsuya Nasukawa at Tokyo Research. We also thank Tony Eng (MIT graduate student) for useful comments on early drafts of this paper, and four other anonymous reviewers whose careful reading greatly improved this paper. Colleagues in business units also provided opportunities for learning about real-world customer requirements. These include Michael Hehenberger, Avijit Chatterjee, Usha Reddy, Edwin Hilpert, Stephen Boyer, John Christina, Peter Libal, Thilo Götz, Thomas Hampp, Deb Sutherland, Lance Snow, Eric Will, Jeff Tenner, Dave Herbeck, Erik Voldal, Bryan Tousley, Joanna Batstone, Beth Smith, and of course, Carol Kovac.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of United States National Library of Medicine, Institut Suisse de Bioinformatique (SIB) Foundation, Intel Corporation, or Linus Torvalds.

Cited references and notes

1. *Pharma2010: The Threshold of innovation*, IBM Corporation (2002), http://www-1.ibm.com/industries/healthcare/doc/content/resource/thought/390030105.html.
2. W. C. Swope, "Deep Computing for the Life Sciences," *IBM Systems Journal* **40,** No. 2, 284–262 (2001), http://www.research.ibm.com/journal/sj/402/swope.html.
3. J. Augen, "The Evolving Role of Information Technology in the Drug Discovery Process," *Drug Discovery Today* **7,** No. 5, 315–323 (March 2002).
4. D. Ferrucci and A. Lally, "Building an Example Application with the Unstructured Information Management Architecture," *IBM Systems Journal* **43,** No. 3, 455–475 (2004, this issue).
5. D. Ferrucci and A. Lally, "Accelerating Corporate Research in the Development, Application and Deployment of Human Language Technologies," *Proceedings of the Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, Edmonton, CA (May 31, 2003).
6. A. D. Marwick, "Knowledge Management Technology," *IBM Systems Journal* **40,** No. 4, 814–830 (2001).
7. R. Mack, R. Byrd, and Y. Ravin, "Knowledge Portals and the Emerging Digital Knowledge Workplace," *IBM Systems Journal* **40,** No. 4, 925–955 (2001).
8. D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, Saddle River, NJ (2000).
9. R. Baeza-Yates and B. Ribeiro-Neto, B. (Editors) *Modern Information Retrieval*, ACM Press, New York (1999).
10. C. Blaschke, L. Hirschman, and A. Valencia, "Information Extraction in Molecular Biology," *Briefings in Bioinformatics* **3,** No. 2, 154–165 (June 2002).
11. J. Tsujii, "Tutorial on Information Extraction in Biological Sciences," *Proceedings of the Pacific Symposium on Biocomputing* (2001), http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/tutorial/index.html#psb2001.
12. L. Hunter, *On-line Course Notes on Bioinformatics*, Center for Computational Pharmacology, University of Colorado Health Sciences Center, http://compbio.uchsc.edu/hunter/.
13. T. Nasukawa and T. Nagano, "Text Analysis and Knowledge Mining System," *IBM Systems Journal* **40,** No. 4, 967–984 (2001).
14. W. Cody, J. Kreulen, V. Krishna, and W. S. Spangler, "The Integration of Business Intelligence and Knowledge Management," *IBM Systems Journal* **41,** No. 4, 697–713 (2002).
15. J. Chang, S. Raychaudhuri, and R. Alman, "Including Biological Literature Improves Homology Search," *Proceedings of the Pacific Symposium on Biocomputing*, World Scientific, River Edge, NJ, 374–383 (2001).
16. S.-K. Ng and M. Wong, "Toward Routine Automatic Pathway Discovery from Online Scientific Text Abstracts," *Genome Informatics* **10,** 104–112 (1999).
17. T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated Extraction of Information on Protein-protein Interactions from the Biological Literature," *Bioinformatics* **17,** 155–161 (2001).
18. T.-K. Jenssen, A.-L. Komorowski, and E. Hovig, "A Literature Network of Human Genes for High Throughput Analysis of Gene Expression," *Nature Genetics* **28,** 21–28 (May 2001).
19. T. Huynh, I. Rigoutsos, L. Parida, D. Platt, and T. Shibuya, "The Web Server of IBM's Bioinformatics and Pattern Discovery Group," *Nucleic Acids Research* **31,** No. 13, 3645–3650 (2003).
20. IBM Research, Computational Biology Center, http://www.research.ibm.com/bioinformatics/.
21. Swiss-Prot is available (along with other related databases) at http://www.expasy.org/sprot/.
22. *Life Sciences Framework*, IBM Corporation (October, 2003), http://www-3.ibm.com/solutions/lifesciences/solutions/framework.html.
23. *DB2 Information Integrator for Content*, IBM Corporation, http://www-3.ibm.com/software/data/eip/.
24. *IBM LanguageWare Linguistic Engine*, http://www-306.ibm.com/software/globalization/topics/languageware/index.jsp.
25. The LanguageWare linguistic engine provides dictionaries for several languages. A legacy version, LanguageWare v2,

was embedded in the pre-UIMA Textract tool, available in the IM4T toolkit.
26. *Intelligent Miner for Text Product Review*, IBM Corporation (February, 1997), http://www-3.ibm.com/software/data/iminer/fortext/.
27. M. Neff, R. Byrd, and B. Boguraev, "The Talent System: TEXTRACT Architecture and Data Model," *Proceedings of the Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, Edmonton, CA (May 31, 2003).
28. The default training corpus used to train the POS tagger is available from the Language Data Consortium (http://www.ldc.upenn.edu/) and is based on non-biomedical text content and style.
29. "IBM, SAS join to help automakers to comply with TREAD act," *Computer World* (April 3, 2003).
30. M. McCord, "Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars," in *Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science*, R. Studer Editor, Springer Verlag, Berlin (1990), pp. 118–145.
31. Unified Medical Language System (UMLS), http://www.nlm.nih.gov/research/umis.
32. LocusLink, http://www.ncbi.nlm.nih.gov/LocusLink/.
33. National Library of Medicine, http://www.nlm.nih.gov/libserv.html.
34. L. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. Batra, P. Kamesam, and R. Kothari, "Information Extraction from Biomedical Literature: Methodology, Evaluation and an Application," *Proceedings of the 2003 ACM CIKM Conference*, New Orleans, LA (2003).
35. International Union of Pure and Applied Chemistry, http://www.chem.qmw.ac.uk/iupac/.
36. WEKA Machine Learning Project, http://www.cs.waikato.ac.nz/~ml/.
37. Personal communication, David E. Johnson, IBM Thomas J. Watson Research Center.
38. T. Zhang, F. Damereau, and D. E. Johnson, "Text Chunking Based on a Generalization of Winnow," *Journal of Machine Learning Research* **2,** 615–637 (2002).
39. T. Zhang, "Regularized Winnow Methods," *Advances in Neural Information Processing Systems* **13,** 703–709 (2001).
40. J. Cooper and R. Byrd, "Lexical Navigation: Visually Prompted Query Expansion and Refinement," *Proceedings of ACM DIGLIB97*, Philadelphia, PA (1997), pp. 237–246.
41. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics* **25,** 25–29 (2000).
42. E. Brown, A. Dolbey, and L. Hunter, "IBM Research and the University of Colorado TREC 2003 Genomics Track," *Proceedings of the 12th NIST TREC Conference* (November, 2003), http://trec.nist.gov/pubs/trec12/papers/ibm-brown.genomics.pdf.
43. A. Inokuchi and H. Kashima, "Mining Significant Pairs of Patterns from Graph Structures with Class Labels," *Proceedings of the Third IEEE International Conference on Data Mining*, Melbourne, FL (November 2003).
44. The GENIA project Web site provides links to several resources of Junichi Tsujii, http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.
45. These three measures are standard measures of accuracy in finding specified categories of objects in search or information extraction. "Recall" is the percentage of correct identifications relative to all possible correct answers in the collection. "Precision" is the percentage of correct identification relative to all correct answers provided by the system. The F-value is derived from and tries to balance the trade-off implied in precision and recall measures (see Reference 8, page 578).
46. P. Youngja and R. Byrd, "Hybrid Text Mining for Finding Terms and Their Abbreviations," *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-2001)*, Carnegie Mellon University, Pittsburgh, PA (June 2001), http://www.cs.cornell.edu/home/llee/emnlp.html.
47. Munich Information Center for Protein Sequences, http://www.mips.biochem.mpg.de/.
48. J. Cooper, "An Evaluation of Unnamed Relations in Discovery of Protein-Protein Interactions," Presented at ACM SIGIR 2003, *Workshop on Text Analysis and Search for Bioinformatics*, Toronto, CA (2003).
49. C. Blaschke, M. Andrade, C. Ouzounis, and A. Valencia, "Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions," *Proceedings of the International Conference on Intelligent Systems in Molecular Biology* (1999), pp. 60–67.
50. T. Rindflesch, L. Tanabe, J. Weinstein, and L. Hunter, "EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature," *Proceedings of the 5th Pacific Symposium on Biocomputing* (2000), pp. 538–549.
51. J. Sowa, "*Conceptual Structures: Information Processing in Mind and Machine. Reading,"* Addison-Wesley, Reading, MA (1984).
52. M. Gruninger and J. Lee, "Ontology Applications and Design, Introduction," *ACM Communications* **45,** No. 2, 39–41 (February 2002).
53. B. Humphreys, D. Lindberg, H. Schoolman, and G. Barnett, "The Unified Medical Language System: An Information Research Collaboration," *Journal of the American Medical Informatics Association* **5,** 1–11, (1998).
54. D. Carmel, E. Amitay, M. Herscovici, Y. Maarek, Y. Petruschka, and A. Soffer, "Juru at TREC 10 - Experiments with Index Pruning," *Proceedings of the 10th Text Retrieval Conference* (2001).
55. D. Carmel, Y. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer, "Searching XML Documents via XML Fragments," *Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, Toronto, Canada, (August 2003), pp. 151–158.
56. The Juru text search engine is described in http://www.haifa.il.ibm.com/km/ir/juru/ and the version of Juru for indexing and searching XML fields is described in "Juru XML - an XML retrieval system at INEX'02," http://qmir.dcs.qmul.ac.uk/inex/Slides/YosiMass_etal_talk.pdf.
57. R. Y. Ando, B. Boguraev, R. Byrd, and M. Neff, "Multidocument Summarization by Visualizing Topic Content," *Proceedings of ANLP/NAACL Workshop on Automatic Summarization*, Seattle, WA (2000), pp. 79–88.
58. S. Vaithyanathan and B. Dom, "Model-Based Hierarchical Clustering," *Proceedings of UAI-2000*, Stanford (2000), http://citeseer.nj.nec.com/vaithyanathan00modelbased.html.
59. K. Krishna and R. Krishnapuram, "A Clustering Algorithm for Asymmetrically Related Data with its Applications to Text Mining," *Proceedings of the 2001 ACM CIKM Conference*, Atlanta, GA, (2001), pp. 571–573.
60. N. Uramoto, H. Matsuzawa, T. Nagano, A. Murakami, H. Takeuchi, and K. Takeda, "A Text-Mining System for Knowledge Discovery from Biomedical Documents," *IBM Systems Journal* **43,** No. 3, 516–533 (2004, this issue).

61. S. Grell, "Information Retrieval in Life Sciences: How to Discover Relations between Concepts," Unpublished master's thesis, University of Heidelberg (December, 2001).
62. R. Mack and M. Hehenberger, "Text-Based Knowledge Discovery: Search and Mining of Life-Sciences Documents," *Drug Discovery Today*, **7,** 89–98 (2002).
63. D. Swanson, "Implicit Text Linkages between MEDLINE Records: Using Arrowsmith as an Aid to Scientific Discovery," *Library Trends* **48,** No. 1, 48–59 (1999).
64. M. Weeber, H. Klein, T. W. Lolkje, and L. de Jong-van den Berg, "Using Concepts in Literature-Based Discovery: Simulating Swanson's Raynaud-Fish Oil and Migraine-Magnesium Discoveries," *Journal of the American Society for Information Science and Technology* **52,** No. 7, 548–557 (2001).
65. P. Kankar, S. Adak, A. Sarkar, K. Murari, and G. Sharma, "MedMeSH Summarizer: Text Mining for Gene Clusters," *Proceedings of the Second SIAM International Conference on Data Mining*, Arlington, VA, (2002), pp. 548–565.

Accepted for publication April 18, 2004.

Robert Mack *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, N.Y., 10532 (robertmack@us.ibm.com).* Dr. Mack is a research staff member at the IBM Watson Research lab. He received his Ph.D. degree in cognitive and experimental psychology at the University of Michigan in 1981. He has worked on a wide range of software systems, both as a user-interface and human-computer-interaction specialist, and as a project leader for prototype systems and middleware for applications relating to digital libraries, digital video archive and search, and knowledge portals. Dr. Mack and his team are currently working on applying text mining to support knowledge discovery in life science.

Sougata Mukherjea *IBM Research Division, India Research Lab., Block 1, IIT Hauz Khas, New Delhi, India 110017 (smukherj@in.ibm.com).* Dr. Mukherjea is a research staff member at the IBM India Research Lab. He received a bachelor's degree from Jadavpur University in Calcutta, a master's degree from Northeastern University in Boston, and a Ph.D. degree from the Georgia Institute of Technology in Atlanta (all in computer science). Before joining IBM, he held research and software architecture positions in companies in the Silicon Valley (California) including NEC USA, BEA Systems, and Verity. His research interests include information visualization and retrieval and applications of text mining in areas such as Web search and bioinformatics.

Aya Soffer *IBM Research Division, Haifa Research Laboratory, University Campus, Haifa, Israel 31905 (ayas@il.ibm.com).* Dr. Soffer is a research staff member at the IBM Research Lab in Haifa, Israel, where she manages the Information Retrieval group. She received her M.S. and Ph.D. degrees in computer science from the University of Maryland at College Park in 1992 and 1995, respectively. Before joining IBM, Dr. Soffer was a research scientist at the NASA Goddard Space Flight Center, where she worked on digital libraries for earth science data. Her research interests include information retrieval, pictorial information systems, multidimensional indexing, and non-traditional database systems. Dr. Soffer and her group have developed several knowledge management solutions for IBM and Lotus products, among them the WebSphere* Portal's search engine.

Naohiko Uramoto *IBM Research Division, Tokyo Research Laboratory, 1623-14, Shimotsuruma, Yamato, Kanagawa, Japan (uramoto@jp.ibm.com).* Dr. Uramoto joined the IBM Tokyo Research Laboratory in 1990 after receiving a master's degree in computer science from Kyushu University. He received a Ph.D. degree in computer science from Kyushu University in 1990. He has been a visiting associate professor at the National Institute of Informatics since 2000. His research interests include natural language processing, text mining, XML and meta-data, and information integration.

Eric Brown *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, N.Y., 10532 (ewb@us.ibm.com).* Dr. Brown is a research staff member at the Watson Research Center. He received a B.S. degree from the University of Vermont and M.S. and Ph.D. degrees from the University of Massachusetts at Amherst, all in computer science. Dr. Brown joined IBM in 1995 and conducts research in a variety of information-retrieval and text-analysis areas, including parallel and distributed information retrieval, question answering, text categorization, applications of speech recognition in knowledge management, and text analysis and search for biomedical text.

Anni Coden *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, N.Y., 10532 (anni@us.ibm.com).* Dr. Coden is a research staff member at the Watson Research Center. She received her Ph.D. degree in computer science from the Massachusetts Institute of Technology in 1981 and continued there as a research scientist. She joined the IBM Research Division in 1982, and has worked in various areas of computer science. She started her career in developing tools for computer chip design and continued her work in IBM's antivirus project. More recently, she worked on algorithms and systems in the area of digital multimedia libraries and on analyzing and searching speech transcripts in real-time video systems. Her current investigations are in the area of automatically analyzing medical records for knowledge discovery.

James Cooper *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, N.Y., 10532 (jwcnmr@us.ibm.com).* Dr. Cooper is a research staff member at the Watson Research Center, where he studies problems in text analytics and object-oriented design. He received his Ph.D. degree in chemistry and has been working on problems in computing and chemistry at IBM and elsewhere in the industry for more than 20 years. He is a well-known writer, having published 15 books in computer science, is a columnist for JavaPro Magazine, and is currently the most widely translated author at IBM Research.

Akihiro Inokuchi *IBM Research Division, Tokyo Research Laboratory, 1623-14, Shimotsuruma, Yamato, Kanagawa, Japan (inokuchi@jp.ibm.com).* Dr. Inokuchi joined the IBM Tokyo Research Laboratory in 2000 after receiving a master's degree in communication engineering from Osaka University. He received a Ph.D. degree in communication engineering from Osaka University in 2004. His current focus is on data mining, machine learning, text mining, chemo-informatics, bioinformatics, and their applications.

Bhavani Iyer *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, N.Y., 10532 (bsiyer@us.ibm.com).* Ms. Iyer is a software engineer at the Watson Research Center. She received an M.B.A. degree in finance from the Asian Institute of Management in the Philippines and an M.S. degree in computer science and information management from Pace University. Her most recent project involved the application of text analytics and unstructured information management systems to the life-science domain. Other interests include database management, distributed systems, data mining, and systems and application integration.

Yosi Mass *IBM Research Division, Haifa Research Laboratory, University Campus, Haifa, Israel 31905 (yosimass@il.ibm.com).* Mr. Mass received a B.Sc. degree in computer science from Bar Ilan University in Israel in 1980 and an M.Sc. degree from the Weizmann Institute in Israel in 1993. Mr. Mass has worked for IBM since 1996 at the Haifa Research lab as a researcher. Before joining IBM he worked in the Israeli high-tech industry for Orbotech and for Ubique. His research areas include virtual reality, visualization, security and trust, multi-user collaboration, XML, and information retrieval. In recent years, he has been working on an information retrieval engine for search in XML collections.

Hirofumi Matsuzawa *IBM Research Division, Tokyo Research Laboratory, 1623-14, Shimotsuruma, Yamato, Kanagawa, Japan (matuzawa@jp.ibm.com).* Mr. Matsuzawa joined the IBM Tokyo Research Laboratory in 1993 after receiving a master's degree in software engineering from Waseda University. His experience includes 3D solid modeling, business intelligence, parallel OLAP, data mining, and text mining. His current focus is on data mining, text mining, and the application of grid architectures to life science.

L. Venkata Subramaniam *IBM Research Division, India Research Lab., Block 1, IIT Hauz Khas, New Delhi, India 110017 (lvsubram@in.ibm.com).* Dr. Subramaniam has been a research staff member at the IBM India Research lab since 1998. He received a B.E. degree in electronics and communications engineering from the University of Mysore in India, an M.S. degree in electrical engineering from Washington University in St. Louis, and a Ph.D. degree in electronics from the Indian Institute of Technology in New Delhi, in 1991, 1993, and 1998, respectively. His current research interests include statistical natural language processing, text and data mining, and bioinformatics.