Preface

Medical informatics is an emerging interdisciplinary science that deals with clinical health-related information, its structure, acquisition and use. Medical Informatics (MI) is grounded in the principles of computer science, artificial intelligence, as well as the clinical and basic sciences. MI includes scientific endeavors ranging from building theoretical models and evaluation of applied systems.

Computational intelligence is a well-established paradigm, where new theories with a sound biological understanding have been evolving. Defining computational intelligence is not an easy task. In a nutshell, which becomes quite apparent in light of the current research pursuits, the area is heterogeneous with a combination of such technologies as neural networks, fuzzy systems, rough set, evolutionary computation, swarm intelligence, probabilistic reasoning, multi-agent systems etc. The recent trend is to integrate different components to take advantage of complementary features and to develop a synergistic system.

This book deals with the application of computational intelligence in medical informatics. Addressing the various issues of medical informatics using different computational intelligence approaches is the novelty of this edited volume. This volume comprises of 15 chapters' including an introductory chapter giving the fundamental definitions and some important research challenges. Chapters were selected on the basis of fundamental ideas/concepts rather than the thoroughness of techniques deployed. The fifteen chapters are organized as follows.

In the introductory Chapter, *Kelemen et al.* provide a review of recently developed theories and applications in computational intelligence for gene-gene and gene-environment interactions in complex diseases in genetic association study.

Chapter 2 by *Burns et al.* is designed to act as an introduction to the field of biomedical text-mining for computer scientists who are unfamiliar with the way that biomedical research uses the literature. Authors describe how creating knowledge bases from the primary biomedical literature is formally equivalent to the process of performing a literature review or a 'research synthesis'. The main body of the chapter is concerned with the use of text mining approaches to populate knowledge representations for different types of experiment. Authors provide a detailed example from neuroscience and describe a detailed description of the methodology used to perform the text mining based on the conditional random fields model.

Tamalika Chaira and *Tridib Chaira* in Chapter 3 propose a new image segmentation technique using intuitionistic fuzzy set and is applied for brain images and blood cell for edge detection. The proposed method works well even on poor quality images.

In Chapter 4, *Kreinovich* and *Shpak* illustrate that in general, detecting aggregability is NP-hard even for linear systems, and thus (unless P=NP), there is only hope to find efficient detection algorithms for specific classes of systems. Authors illustrate that in the linear case, once the blocks are known, it is possible to efficiently find appropriate linear combinations.

Siebel et al. in the fifth Chapter propose the automatic design of neural networks as a controller in a visuo-motor control scenario. Evolutionary Acquisition of Neural Topologies (EANT) uses evolutionary search methods on two levels: In an outer optimization loop called structural exploration new networks are developed by gradually adding new structures to an initially minimal network. In an inner optimization loop called structural exploitation the parameters of current networks are optimized. EANT was used with a complete simulation of a visuo-motor control scenario to learn neural networks by reinforcement learning.

In Chapter 6, *Lee at al.* propose Block Principal Components Analysis (PCA) and a variable selection method based on principal component loadings for dimension reduction. The main focus of this is how to deal with large number of variables (gene expressions) in microarray data sets. Authors also investigate the effect ill-conditioning has on the Mahalanobis distance between clusters using the well-known Hilbert matrix.

De Roberto Jr et al. in the seventh Chapter describes the development of a new tool for genome interpretation. The software recognizes coding regions with a user-friendly interface. The system is based on a gene model and combines the weight-position matrix technique with the flexibility of artificial neural networks in classification problems.

In Chapter 8, *Bogan-Marta et al.* discuss about the diversity of language engineering techniques and those involving information theoretic principles in analyzing protein sequences from similarity perspective. Authors also present a survey of the different approaches identified with a focus on two methods, which the they experimented.

Sehgal et al. in Chapter 9, investigate the impact of missing values on post genomic knowledge discovery methods like, gene selection and Gene Regulatory Network (GRN) reconstruction. A framework for robust subsequent biological knowledge inference is proposed, which has shown significant improvements in the outcomes of gene selection and GRN reconstruction methods.

In Chapter 10, *Sehgal et al.* provide a comprehensive comparative study on GRN reconstruction algorithms. The methods discussed are diverse and vary from simple similarity based methods to state of the art hybrid and probabilistic methods. The Chapter also emphasizes the need of strategies, which should be able to model the stochastic behavior of gene regulation in the presence of limited number of samples, noisy data, multi-collinearity for high number of genes.

Kasabov et al. in Chapter 11 present some preliminary results on the Brain-Gene Ontology project that is concerned with the collection, presentation and use of knowledge in the form of ontology equipped with the knowledge discovery means of computational intelligence. Brain-Gene Ontology system includes various concepts, facts, data, graphs, visualizations, animations, and other information forms, related to brain functions, brain diseases, their genetic basis and the relationship between all of them, and various software simulators.

In Chapter 12, *Dohnal et al.* present the metric space approach and its applications in the field of Bioinformatics. Authors describe some of the most popular centralized disk-based metric indexes with a focus on parallel and distributed access methods, which can deal with data collections that for practical purposes can be arbitrary large. An experimental evaluation of the presented distributed approaches on real-life data sets is also presented.

Kroc in Chapter 13 illustrates that the development of an adequate mechanical model of living tissues provides the morphological model with sufficient flexibility necessary to achieve expected morphological development. Author focuses on the development of mesenchymal and epithelial tissues, which creates the basic mechanism of tooth development.

In Chapter 14, *Bobrik et al.* describe two simplified models of Darwinian evolution at the molecular level by applying the methods of artificial chemistry. A metaphor of a chemical reactor (chemostat) is considered and then a simplified formal system *Typogenetics*, is discussed.

In the last Chapter *Bobrik et al.* present the third simplified model of Darwinian evolution at the molecular level, following the two presented in Chapter 14. An artificial-life application is designed as a modification of the metaphor of chemostat, where the secondary structure of binary strings specifies instructions for replication of binary strings.

We are very much grateful to the authors of this volume and to the reviewers for their tremendous service by critically reviewing the chapters. The editors would like to thank Dr. Thomas Ditzinger (Springer Engineering Inhouse Editor) and Professor Janusz Kacprzyk (Editor-in-Chief, Springer Studies in Computational Intelligence Series) and Ms. Heather King (Springer Verlag, Heidelberg) for the editorial assistance and excellent cooperative collaboration to produce this important scientific work. We hope that the reader will share our excitement to present this volume on '*Computational Intelligence in Medical Informatics*' and will find it useful.

Arpad Kelemen, Ajith Abraham and Yulan Liang (Editors)
August 2007

Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples

Gully A.P.C. Burns, Donghui Feng, and Eduard Hovy

Information Sciences Institute / USC, 4676 Admiralty Way Suite 1001 Marina del Rey, CA 90292 {burns,donghui,hovy}@isi.edu

Summary In this chapter, we describe how creating knowledge bases from the primary biomedical literature is formally equivalent to the process of performing a literature review or a 'research synthesis'. We describe a principled approach to partitioning the research literature according to the different types of experiments performed by researchers and how knowledge engineering approaches must be carefully employed to model knowledge from different types of experiment. The main body of the chapter is concerned with the use of text mining approaches to populate knowledge representations for different types of experiment. We provide a detailed example from neuroscience (based on anatomical tract-tracing experiments) and provide a detailed description of the methodology used to perform the text mining itself (based on the Conditional Random Fields model). Finally, we present data from text-mining experiments that illustrate the use of these methods in a real example. This chapter is designed to act as an introduction to the field of biomedical text-mining for computer scientists who are unfamiliar with the way that biomedical research uses the literature.

2.1 Introduction

The overwhelming amount of information available to the biomedical researcher makes the use of intelligent computational tools a necessity. These tools would help the researcher locate information appropriate to his or her goal, identify/extract the precise fragments of information required for each specific task, correlate and sort the extracted information as needed, and summarize or otherwise synthesize it in ways suitable for the task at hand. Such tools are not easy to build, by and large, and require expertise in a variety of computer science specialties, including database management, data analysis, natural language processing, and text mining.

Naturally, the organization and nature of the information to be so manipulated has a great influence on the nature and level of performance of the tools used. For example, the bioinformatics systems most widely used by biomedical researchers, those

Gully A.P.C. Burns et al.: *Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples*, Studies in Computational Intelligence (SCI) **85**, 17–50 (2008)
www.springerlink.com

hosted by the National Center for Biotechnology Information (NCBI) [1], include two types of database: (a) molecular and genetic databases and (b) bibliographic databases (notably PubMed and PubMed Central). The structure of information contained in the bioinformatics databases is precisely defined, tabulated, standardized, homogeneous, and concerned mainly with molecular/genetic data and their derivations. In contrast, the information contained in bibliographic databases, defined as it is in natural language, is non-standardized, massively heterogeneous, and concerned with all aspects of biomedical knowledge (physiology and anatomy at all levels of behavior, the body, its constituent organ systems and subdivisions).

The differences between these two types of system provide the central theme of this chapter: while it is relatively straightforward with current computational techniques to master the former, well-organized material, the latter requires sophisticated natural language processing (NLP) techniques. Additionally, the process of using the literature confirms to rigorous scholarly standards that necessitate careful citation of known data, the accurate representation of complex concepts and (in the case of formal meta-analysis) rigorous statistical analysis [2]. The development of Knowledge Bases (KBs) by manual curation of the literature is being used in a great many fields, including neuroanatomy [3–5], and yeast protein interaction networks [6]. The use of NLP techniques has generated large-scale systems such as Textspresso (for fly genetics [7]) and Geneways (for signal transduction [8]), amongst others.

In many sub-disciplines of the life-sciences (such as non-molecular neuroscience, physiology and endocrinology), there are no large scale databases that tabulate experimental findings for researchers to browse and view. In these subjects, the vast amount of scientific information *is only available to the community in the form of natural language*. The impact of this can be seen in the world of difference that exists between neuroinformatics and bioinformatics databases. The CoCoMac system (‘Collations of Connectivity data on the Macaque brain’, [4, 9]) is a successful neuroinformatics database project concerned with inter-area connections in the cerebral cortex of the Macaque. It is a mature solution for a problem that was under consideration by national committees concerned as far back as 1989 (L.W. Swanson, personal communication). CoCoMac currently contains roughly 4×10^4 connection reports from 395 papers and is the product of years of painstaking data curation by its development team. By way of contrast, the National Library of Medicine announced in Aug 2005 that the total quantity of publicly available genetic data was 10^{10} individual base pairs from over 165,000 organisms.

Why is there such a massive disparity (six orders of magnitude) between the two types of system? Two key components are present in molecular bioinformatics systems and absent in the other domains: high-throughput Knowledge Acquisition (KA) methods, and appropriately expressive target Knowledge Representation (KR) systems to hold the experimental findings in a coherent structure. High-throughput data acquisition methods have been developed for molecular work and their outputs are relatively quantitative and simple (in comparison to the heterogeneous complexity of neuroscience data). Databases such as NCBI [1], UniProt [10] and the Kyoto Encyclopedia of Genes and Genomes (KEGG [11]) are the products of the use of this technology for over a decade or more. If high throughput knowledge acquisition methods could be used on the published literature to populate a representation that captures the essential details and linkage of experiments, the size of databases such as CoCoMac could increase significantly.

In this chapter, our objective is to describe high-throughput methods to construct KBs based on the application of NLP to the primary experimental literature. A major point of discussion for this work is the ontology engineering methodolgy used to design the target KR that we are attempting to populate. Scientifically speaking, the process of generating a KB in this way is equivalent to the task of compiling a literature review, or more formally: 'research synthesis' [12]. Given the large quantities of effort expended by biomedical scientists studying the literature, we suggest that study tools could have a large impact on the field [13, 14].

Ontology engineering and development is attracting much interest within the Biomedical Informatics community. A National Center for Biomedical Ontology (NCBO) has been established at Stanford University [15], with support from several similar research teams. Ontologies may be defined as 'specifications of conceptualizations' [16], and work in this field is mature, supported by thirty years of research into Artifial Intelligence (AI), widely used data standards (OWL and RDF, Common Logic, *etc.*), codes of best practice (*e.g.,* the Open Biomedical Ontology foundry: http://obofoundry.org/), and an increasing number of reliable open-source software systems [17–19].

This chapter is therefore designed to serve a dual purpose: to provide a philosophical context of text mining work and to describe the process in concrete, experimental terms. We begin by introducing the idea of 'conceptual biology' in section 2, and how this relates to text mining in general. In section 3, we then describe the rationale for partitioning the primary experimental literature based on 'experimental type' and how this provides structure for text-mining work. We discuss existing biomedical knowledge bases that have been derived from the literature in section 4, and then describe how the process of preparing a review article can define the underlying context of the knowledge-intensive task that we are addressing in section 5. In the latter half of the chapter, we show an example from neuroscience by elaborating the theory (section 6), methodology (section 7) and results (section 8) of text-mining experimental work for a specific example taken from neuroscience that we introduced in earlier sections. Finally we synthesize this work as a vision for the development of the next generation of biomedical informatics systems.

2.2 A framework for conceptual biology

The term 'conceptual biology' denotes a computational approach that is based on synthesizing new knowledge from data found in existing, already-published work [20, 21]. Although, this approach lies at the heart of all literature-driven bioinformatics systems, the term is itself rarely used explicitly. The originators of the idea of conceptual biology developed the ARROWSMITH tool [22].

Biology is intrinsically concept-driven and is massively heterogeneous with respect to the representations used for different concepts. This heterogeneity is simply an emergent property of the way that biological experiments are performed and synthesized into facts. The Gene Ontology illustrates this by describing over 21,000 separate terms describing biological processes, cellular components and molecular functions [23].

Let us begin by asking the following question: *How do experiments contribute to new knowledge that may be used by other researchers?*

In Chapter 3 of [24], a non-biologist describes this process from 1962 to 1969 for a Nobel-Prize-winning discovery: the chemical structure of the hormone Thyrotropin Releasing Factor (TRF). This process was multi-stage, iterative and evolutionary. Firstly, it was recognized that the discovery of a new fact (the structure of TRF) would provide an important breakthrough, and then a research community went about solving the problem through a range of experimental techniques. The members of this community were competing fiercely with each other, so that the discoveries of one group were immediately used by the others. Finally, in 1969, the discovery was made that the chemical formula of TRF is 'Pyro-Glu-His-Pro-NH2'. It is important to note that up until that point, publications from the members of the community were concerned with arriving at the definition of the chemical formula. When the formula was derived, all of the previous work was summarized by this single piece of information, which was then available for use by other scientists without any reference to the research that had lead to its discovery.

The research process therefore generates a plethora of different complex representations, arguments and data to supports the discovery of 'scientific facts'. Such facts are rarely as cut and dried as a chemical formula (as was the case with TRF). They must usually be qualified in terms of supporting evidence and may evolve as more research is conducted.

Knowledge engineering approaches to ontology development seek to formalize this information within a compuatable framework to make it more tractable by the scientific community. The fields of biomedical informatics and computational biology depend on this process of formalization and a number of structured representations are being constructed for this purpose. The OBO foundry lists a number of these ontologies defined at various levels. In typical 'top-level' ontologies, concepts are completely generic (*e.g.,* 'Thing', 'Organism', 'Continuant'). These may be used to inform high-level biomedical concepts such as 'disease', 'stress', 'fear', or 'cancer'. In order to investigate these high-level biomedical concepts, specific **experimental models** are used by scientists to provide a framework to investigate phenomena of interest. These experiments provide a rigorous, logical framework for reasoning about biological phenomena. Principles of experimental design provide the building blocks for this framework and we refer interested computer scientists to [25] for an excellent introduction.

Put simply, biomedical experiments generally consist of the demonstration of statistically significant differences between measurements of a *dependent variable* under conditions imposed by the choice of different values of *independent variables*. This is illustrated schematically in Figure 2.1. Thus, reasoning within biology depends on human expertise, experimental design, statistical inference and

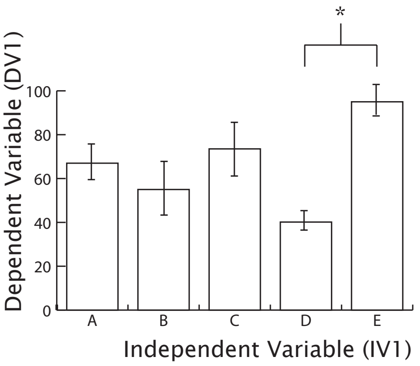


Fig. 2.1. A schematic representation of a hypothetical 'effect' in a biological experiment. The graph shows a statistically significant difference in the dependent variable between conditions D and E.

significance-testing rather than other forms of mathematical logic (which may provide formal methods to model this specialized process).

The values of both independent and dependent variables can be classified based on the four scales of measurement [26]. These scales are (a) ratio measurements (fully numerical with a defined zero point, such as the Kelvin scale of temperature); (b) interval measurements (fully numerical without a defined zero point, such as the Celsius scale); (c) ordinal measurements (ranked data where only relative order is available); (d) nominal measurements (simple enumerated categories). This approach differs from knowledge-engineering approaches where inference rules must be based on boolean values and constructed only from enumerated values.

Thus, a basic individual 'unit' of scientific reasoning is an **experiment**, consisting of a set of independent variables (including the experimental protocol), dependent variables, data and statistics [27]. Typically, conclusions are inferred from this data and then presented as factual statements that are supported by experimental evidence. These facts may then be summarized into a 'model'. This model may then be used as the basis of forming new hypotheses and designing new experiments. This conceptual workflow forms the basis of our approach to text-mining and is illustrated in Figure 2.2.

Within Figure 2.2, we illustrate the construction of knowledge bases from the published literature, raw data files such as images, electronic laboratory notebooks, general multimedia files and online data sources. Any KB would need to store and manage these resources prior to constructing any representation of their content. Descriptions of experimental observations could then be derived either by manual curation, information extraction or other methods. In some cases, facts are also likely to be mined directly from the literature (as is the case with systems that describe protein-protein interactions for example [8]) as well as being inferred from experimental observations.

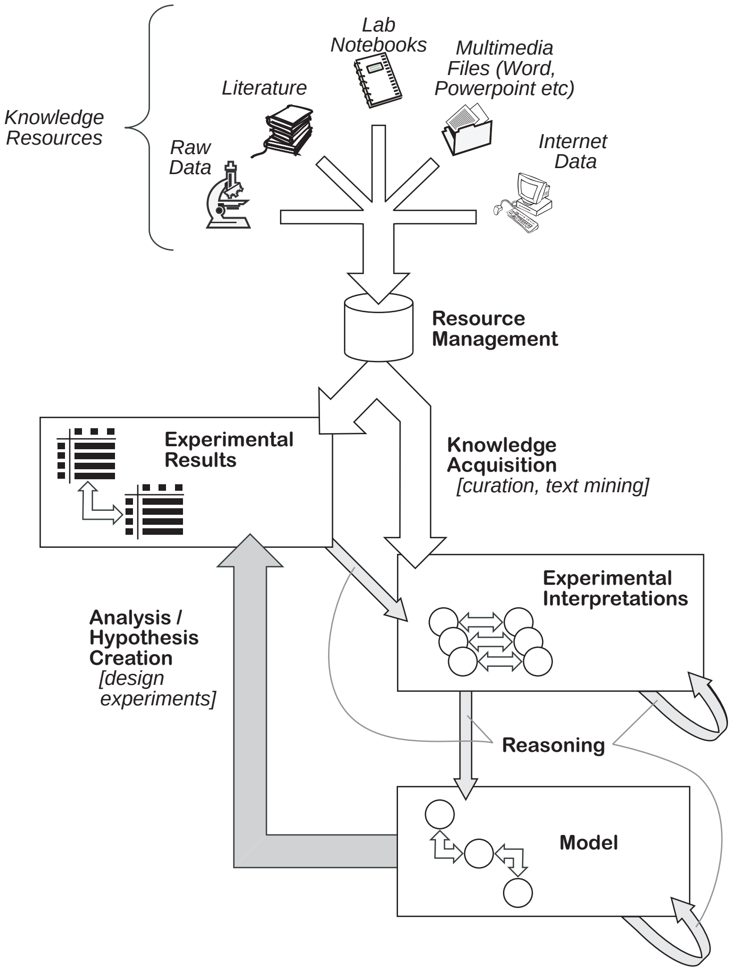


Fig. 2.2. High-level schematic representation for knowledge-base development for conceptual biology based on common knowledge resources. Within this chapter, we emphasize text-mining approaches based on the peer-reviewed published literature, but this framework could conceivably apply to KRs based on any knowledge source.

Standardization is a crucial component of this work. Standards arise from either *de-facto* standard approaches and technology (such as the Gene Ontology [23], or the Protégé system [28]) or from work performed by committees and consortia to agree on appropriate semantics to place on nomenclature and models (such as the Microarray Gene Expression Data Society, or ‘MGED’ or the International Union of Basic and Clinical Pharmacology or ‘IUPHAR’). One viewpoint is that ontologies, data exchange formats, and database schemata constitute ‘computational symbolic theories’ [29]. Certainly, these are the components where the explicit semantics of biology are embedded into technology for use by the community.

In our attempts to construct a framework for conceptual biology, we emphasize the primary experimental observations as the components that are the most portable between different subject domains. For example, descriptions of histological labeling within tract-tracing experiments are almost indistinguishable from descriptions of histological labeling in *in-situ* hybridization experiments; a single represetation could be used for both types of experiment. Experimental observations typically provide accurate assertions, whereas interpretations are dependent on the evidence that support them. We suggest that observations may therefore be more accurate than interpretations within a KB. The drawback of emphasizing observations over interpretations is that additional inference is required to reconstruct the conclusions of a study.

In order to satisfy conditions of exemplary scholarly practise, it is crucial that the KB provide a fully-annotated link to the original phrase, sentence, figure or article section of the mined or curated information (rather than just the citation). These explicit links to text within the source documents and original experimental data enable users to trace the logic that supports the claim that the interpretations are correct. Without these features in place, users of a KB would have to read the paper in its entirety in order to validate its claims.

Given this general architecture, we now examine how we can segment the prohibitively large literature into more manageable domains upon which we may then operate with NLP-based approaches.

2.3 Partitioning the literature - the notion of ‘experimental type’

There are two main types of scientific article: primary experimental reports and reviews. The structure of experimental reports is quite regular and typically has the following sections: abstract, introduction, materials and methods, results, discussion/ conclusion and references (as well as figures and tables scattered throughout). In comparison, the structure of review articles is freeform, and is based mainly on citations linking to knowledge found in experimental reports (or other reviews). We focus on the originating source of new scientific knowledge by only considering primary research articles and disregarding reviews.

Computationally, the literature itself is difficult to partition cleanly, since it is primarily a resource designed with human readability and retrieval in mind. Papers are not separated into information-specific categories that then may be collated into appropriate knowledge bases. To assist with this, we define the notion of **experimental-type** based on the design of an experiment. Although there is variability within the design of any given experiment, we adopt a simplistic approach. All of the seemingly complex choices made by an experimentalist to select a model system, methodology, assaying technique, time-points and type of experimental subject are concerned with the independent variables and their values. All measurements made within the experiment are just dependent variables and their values.

If we wish to construct a database for scientific data for a specific experimental-type, we first construct the database schema based on the experimentalists’ choice of independent and dependent variables. More specifically, we would want our database design to be widely applicable across a large number of individual experiments and we might ignore the less significant choices made by experimenters. This is typified by the idea of 'minimum information required by an experiment' which denotes the level of detail required in the experimental data to be able to make correct high-level interpretations. This idea has formed the basis of standardized object models for specific experiment types to enable collaboration and data sharing [30–33].

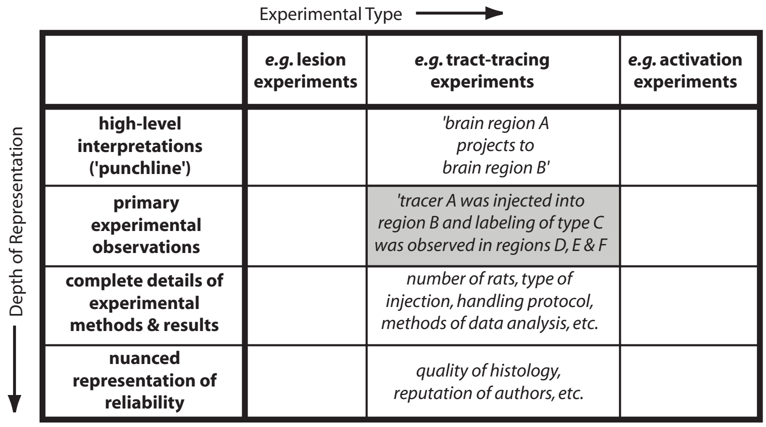
There is a natural parallel between the design of experiments and the design of schemas in bioinformatics systems. Experimental-type is, therefore, a classification of the *knowledge representation schema that can adequately capture the minimum required information for a set of experiments that share the same experimental design and interpretation*. Less formally, if two experiments' data can be entered into a single database in the same set of tables and attributes without needing to alter the database schema, then the two experiments share the same experimental-type.

Another dimension of representation within scientific experiments that we use to partition the literature is the 'depth of representation'. These four categories consist of high-level interpretations and primary experimental observations (shown in Figure 2.2) as well as the complete details of the experimental methods and results (for researchers attempting to replicate the experiment) and a nuanced evaluation of the reliability of the paper. Curation efforts (such as efforts within the GO project [23] and CoCoMac [4]) use codes to denote the source and likely reliability of a specific data entry.

This partition of the primary literature along two orthogonal axes of representation ('experimental-type' and 'depth of representation') is illustrated schematically in Figure 2.3. Within our example described later, we specifically target the shaded 'cell' of this figure: the primary experimental observations of tract-tracing experiments. We will describe tract-tracing experiments later in the chapter. For now, we state that the high-level interpretations of these experiments describe neuronal projections between brain regions and that this information is inferred from experiments where injections of tracer chemicals are made into the brains of experimental animals and then processed to find histological labeling produced by these injections.

It is immediately apparent from Figure 2.3 that two types of text-mining endeavor are possible: 'horizontal' studies that classify papers across experimental type and 'vertical' studies that drill down into the specific literature pertaining to one experimental type. The definition of appropriate knowledge representations and ontologies for each experimental type at each of the different depths of representation is itself a research topic attracting significant interest [31, 34]. Text mining projects that have used unsupervised approaches in biomedicine have been used to index and cluster abstracts in the Medline database [22, 35] provide examples of 'horizontal' studies.

It is possible to identify subtypes and specializations of experiments. Specialized versions of tract-tracing experiments could conceivably include 'double-labeling tract-tracing-experiments', where two (or more) histological labeling methods are used to interactions between neuron populations involved in a projection revealed by the co-localization of labeling. Other examples include ultrastructure experiments (where electron microscopy is used to view the ultrastructure of labeled neurons) and


Experimental Type
Depth of Representation

	***e.g.* lesion experiments**	***e.g.* tract-tracing experiments**	***e.g.* activation experiments**
high-level interpretations ('punchline')		*'brain region A projects to brain region B'*	
primary experimental observations		*'tracer A was injected into region B and labeling of type C was observed in regions D, E & F*	
complete details of experimental methods & results		*number of rats, type of injection, handling protocol, methods of data analysis, etc.*	
nuanced representation of reliability		*quality of histology, reputation of authors, etc.*	

Fig. 2.3. A two-dimensional partition of the published scientific literature.

transneuronal labeling (where tracers may be transmitted between neurons to view multi-synaptic connections) [36]. Each of these experimental types would require a different schema to represent their semantics.

2.4 Related Work: Biomedical Knowledge Bases based on published studies

Thousands of biomedical databases are available to researchers (for a review of the current global state of databases available for molecular biology, see the Molecular Biology Database collection in the 'Database issue' of Nucleic Acids Research [37]). Within the neuroscience community, the 'Neuroscience Database Gateway', provides an online overview of current systems [38]. These systems are often derived from laboratories' primary data (with external links to their publications), rather than synthesizing information found within the literature. Notable systems from within molecular biology that are based on the literature are the BioCyc family of databases (EcoCyc, MetaCyc, *etc.*) [39–41], the BioGRID [42], Textpresso [7], KEGG [43] and GeneWays [8]. The Generic Model Organism Database (GMOD) is a large consortium of organism-specific systems. These include 'dictybase' [44], 'EcoCyc' [39], 'FlyBase' [45], 'MGI' [46], 'RGD' [47], 'SGD' [48], 'TAIR' [49], 'TIGR' [50], 'Wormbase' [51], and 'ZFIN' [52].

A new emerging profession within this field deserves mention: the 'biocurator'. These individuals who populate and maintain large-scale database systems with information in a readable, computable form [53]. As an emerging discipline, biocuration occupies a uniquely important position within biomedical research and this responsibility is often undertaken by teams of Ph.D. level biologists (the Jackson Laboratory has over thirty biocuration staff [54]). Even with this level of commitment and support, most teams are still overwhelmed by the volume of information present in the literature. Crucially, as both databases and ontologies evolve, these large scale efforts will find it increasingly difficult to change the representations they use or update previously curated data to new emerging representations. There is an emerging organized community of biocurators, and the first 'International Biocurator Meeting' was organized in 2005, arising from the collaboration between different databases in the GMOD consortium (http://www.biocurator.org/).

Both Textpresso and GeneWays utilize NLP approaches for knowledge acquisition. Within neuroscience, there are several manually-curated systems: CoCoMac [4] and BAMS [5] describe connections in the Macaque and Rat brain. Also worthy of mention is the work performed within the Tsuji group at the University of Tokyo. Their development of the GENIA text corpus is specifically geared towards the biomedical domain and provides a general reference for Biomedical NLP research [55]. They are also actively engaged in addressed specific research questions concerning information extraction in biomedicine [56].

The BioCyc collection has multiple databases that have undergone extensive manual curation by members of the scientific community [39]. These systems contain knowledge describing signaling pathways derived from (and in conjunction with) genetic data. It has a well-defined ontology [57]. The collection provides model-organism-specific databases (such as EcoCyc for E-Coli [41]) and databases across organisms (such as MetaCyc for metabolism [40]). The EcoCyc database derives information from 8862 publications in the literature (involving extensive manual curation by experts).

The BioGRID system is a large-scale manual curation effort that provides a direct comparison between high-throughput experimental methods for knowledge acquisition and literature curation [6]. It is primarily concerned with protein-protein and protein-gene interactions in yeast (*Saccharomyces cerevisisae*). Within this system, workers have curated information from 31,793 separate abstracts and 6,148 full-text papers [6]. Within this study, the researchers found that the protein interaction datasets taken from high-throughput and literature-curated sources were of roughly equivalent size but only had 14% overlap between them. This suggests that modern, high-throughput techniques augment existing work, but also that the wealth of information reported in the literature cannot easily be reproduced by these methods.

The Textpresso system was originally designed for information retrieval and extraction purposes for the *Caenorhabditis elegans* literature [7]. It involved processing 3,800 full-text articles and 16,000 abstracts and uses regular expression patterns to perform the information extraction step. It also employs entries from the Gene Ontology [23] as entries in its lexicon and used combinatorial rules to build patterns from within programs which could then be applied to its corpus. Systems such as KEGG [43] and the Signal-Transduction Knowledge Environment [58] involves the manual construction of representations of pathways by knowledge engineers.

The GeneWays system is a system for extracting, analyzing, visualizing and integrating molecular pathway data [59]. In 2004, the systems' developers reported that they have downloaded approximately 150,000 articles into the present system. This corpus has yielded roughly 3 million redundant statements that may then be processed with NLP-based approaches [8]. As with the BioCyc systems, GeneWays uses a well defined ontology to describe signaling pathways [59]. Its information extraction system was derived from a mature existing medical Information Extraction (IE) system that is currently used within a production setting in hospitals [60].

It seems clear from this partial review of existing work that the likely coverage of data contained within manually-curated systems is a small percentage of the total that is available. The total size of the published biomedical literature may be estimated from the size of the complete MEDLINE corpus of biomedical citations (available for download as compressed XML from the National Library of Medicine): approximately 1.6×10^7 citations (dating from the mid-fifties to the present day) [61]. Naturally the coverage of each knowledge base may be estimated by the proportion of abstracts available on Medline via a keyword search. If the example presented by the BioGrid is representative, the total number of yeast-specific abstracts was 31,793 and 9,145 were available as full text. Of these, 6,148 papers were curated into the system [6].

Naturally, if the full-text articles are not available, then they simply cannot be included in the contents of a knowledge base. For this reason, licensing and copyright issues become crucial to the development of these systems. Legally, the content of research papers is usually owned by journal publishers. A notable exception to this is the so-called ‘open-access’ publication model. The federal U.S. government has also issued a request that researchers deposit papers that were supported under federal funding into its open-access repository, PubMedCentral [62]. Under normal licensing conditions for most well-funded universities, researchers have access to large numbers of journals as a matter of course. After examining the online availability of journals relevant to the field of neuroendocrinology available at the authors’ home institution (the University of Southern California), we found that from 65 relevant journal titles, we were permitted to access 1886 journal-years worth of text. Thus, even under current conditions of restrictive copyright, it is possible to obtain moderately large quantities of text. Note that, to computer scientists, working in the field of Natural Language Processing, such corpora sizes are not considered large since much work is currently done on terascale crawls of the world wide web [63].

Finally, our own work in this area consists of the NeuroScholar system, which is a knowledge management platform for treatment of the neuroscience literature [13]. This system is designed to be a desktop application that provides direct support for scientists’ interaction with the primary research literature (rather than a single large-scale centralized web-based database system). It provides a means to add both unstructured and structured annotations to full-text articles as PDF files [14], an Electronic Laboratory Notebook component [64] and a system to provide support for visualization plugins based on components such as neuroanatomical atlases [65]. NeuroScholar is open-source and provides a platform for development of knowledge-engineering technology for biomedicine. The system is available for download from http://www.neuroscholar.org/.

In the next section, we examine how the interaction between biologists and computer scientists designing such a system would take into account practices and methods from within the preexisting scholarly process.

2.5 Practical applications: 'Research Synthesis'

The terms 'meta-analysis' and 'research synthesis' refer to formalized approaches to reviewing the literature [12]. Within the clinical, social and behavioral sciences such approaches are widely used to attempt to minimize problems of variance and bias across studies. Due perhaps to the concept-driven nature of the subject, experimental biology does not often rely on these formal approaches directly; instead workers in the field use expert narrative reviews to summarize the knowledge contained in specific subfields of the literature. Thus, we propose that the development of formalized approaches to constructing knowledge bases from the published literature is actually a form of research synthesis or meta-analysis for experimental biology. By leveraging a very large number of studies into this process we seek to (a) increase the possible scope of published reviews and (b) provide tools that make writing conventional reviews easier.

Following [66], the process of constructing a literature review may be broken into stages, where researchers perform specific sets of tasks. We illustrate this workflow in Figure 2.4. Using this approach will involve challenges at each stage which may, or may not, be supported by computational tools.

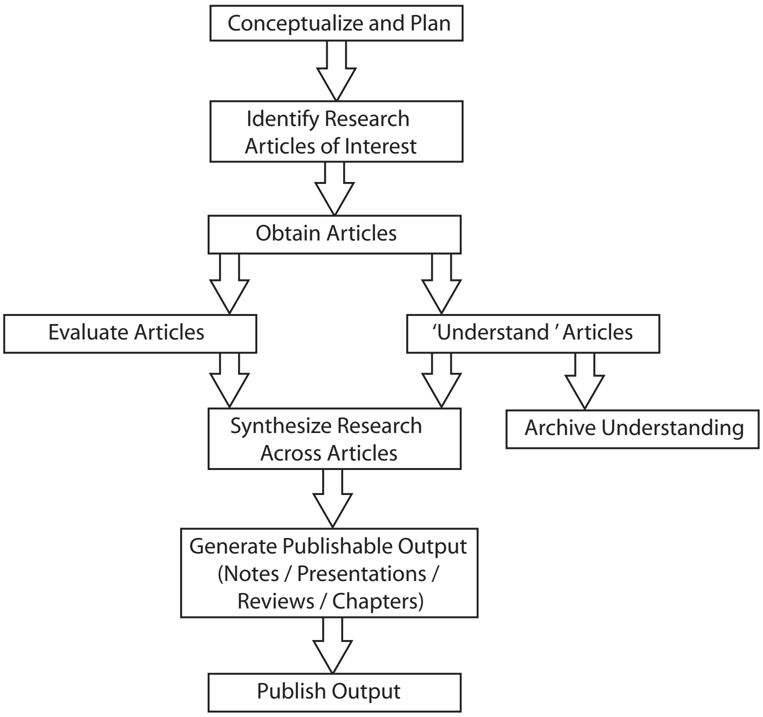

Conceptualize and Plan
Identify Research Articles of Interest
Obtain Articles
Evaluate Articles
'Understand ' Articles
Synthesize Research Across Articles
Archive Understanding
Generate Publishable Output (Notes / Presentations / Reviews / Chapters)
Publish Output

Fig. 2.4. A schematic representation of the process of performing a literature review.

The CommmonKADS framework is a practical methodology of generating solutions for knowledge-intensive tasks [67]. Within this methodology, knowledge engineers are guided through a design process involving a detailed analysis of the task under consideration, the organizational context of the task and the agents performing various roles in relation to it (in addition to issues of representing and acquiring the knowledge necessary to address the task itself). Thus, it is both relevant and necessary to consider the whole task under investigation and to develop models for the contextual and procedural components of the task in question.

At the start of the process of reviewing the literature in a given field, a scientist must first **conceptualize and plan** their study. This currently involves only the researchers' own expertise without support from knowledge tools. Once the researcher has settled on a subject for their review, they must then **identify articles of interest** by searching either Medline or Google Scholar portal. Research is ongoing to improve the performance of these tools (see [68] for an example of adding functionality to Medline). Selecting relevant papers is also called 'literature triage' which has been addressed in community based evaluations such as the KDD challenge cup [69], and TREC 2004 and 2005 [70, 71].

The greatest computationally advances for scientific scholarly work is the ease with which one may now **obtain full-text articles**. This process is determined by copyright issues within the publishing process, and the vast majority of scholarly journals have online access to full-text papers.

The processes of **understanding and evaluating the article** are performed iteratively in parallel depending on how many times and how deeply the researcher reads the paper. Understanding the article may involve reading it several times, taking notes and even discussing the contents of the article with colleagues. Evaluation is based on the quality of the research within the paper (and its relevance to the reviewers). There is not much (if any) computational support for the individual reviewer at this stage, and this work is also the most time consuming and difficult.

Once the researcher has understood the article, he/she may **archive their understanding** by recording notes on file cards, keeping notes, or even storing a structured representation in a local knowledge base [14]. This is an essential component of the process since it is likely that they will forget the details of the article within a few months unless they re-read it. Development within the NeuroScholar system specifically targets this task by providing annotation tools for neuroscientists to be able to store their opinions and accounts as a network of interrelated statements [64]. This is similar to the development of argumentation networks [72, 73]. An important stage of constructing a review is being able to **synthesize research across articles**. This is also the role played by formal meta-analysis when applied across studies with a common set of research criteria (such as with randomized clinical trials [74]). This is a difficult problem, requiring deep knowledge of the subject to create conceptual connections between papers.

Computationally, this is equivalent to data-mining and knowledge discovery. These are analytical techniques that are used extensively within molecular biology to search for correlations and patterns within data stored in databases (see [75–77] for reviews). These are applicable and useful whenever data may be compiled in sufficient quantity, and have also been used in the field of systems-level neuroanatomy to look for patterns in neural connections between brain regions [78, 79].

The final tasks facing the reviewer are to **generate a physical instantiation of the research synthesis** (as the finished review, a presentation or a set of notes) and then **publish or disseminate it**. Given a computational representation, it is relatively simple to generate tabulated or graphical output to represent the knowledge. In addition, Natural Language Generation systems may also be used to create human readable text that summarizes the contents of complex knowledge representations [80].

2.6 An Example Experiment Type: 'Tract-Tracing Experiments'

We describe our approach for a single, specific experiment type: tract-tracing experiments. The methods we describe could be used for any experiment type defined at any depth of representation (see Figure 2.3). This could be accomplished by simply substituting the relevant schema, markup and data into the appropriate place within the methodology.

Tract-tracing experiments were first performed in the early 1970s when tiny amounts of radioactive (tritiated) amino acids were placed into targeted regions of brain tissue [81]. This 'tracer chemical' was taken up by the cells located within the 'injection site' and then transported along the long axonal processes of these neurons. The experimental animal was then sacrificed and its brain processed for radioactivity revealing patches of 'labeling' that revealed the spatial location of the transported tracer. Since these early experiments, the basic design of tract-tracing experiments has remained consistent within the field (see [82] for a treatment of newer methods). The types of tracers used in modern experiments are easier to use, are more precise, suffer less from tissue-specific problems (such as uptake of the tracer by fibers passing through the injection site but not terminating there), and they produce clear histological labeling of cells and their processes. The consistency and relative simplicity of this experimental design, coupled with the number of studies performed and the relative importance and complexity of the resulting data (connections in the brain), sparked the development of several databases of neural connectivity over the last 15 years where the information from these experiments has been partially stored [4, 5, 83, 84]. None of these systems use text mining approaches and all have partial coverage of the literature.

An object-oriented model that captures the logic of this schema is expressed in UML in Figure 2.5, (see also [14, 34]). The logical design of a tract tracing experiment is relatively simple consisting of three sets of entities that may be defined as part of a schema. The first is the **`chemical`** used as a tracer in the experiments since anterograde tracers (such as Phaseolus Leuco-Agglutinin or 'PHAL' [85]) reveal the outputs of the neurons in the injection site, and retrograde tracers (such as Fluoro Gold or 'FG' [86]) reveal the inputs. Thus the uptake properties of each tracer determine how we should interpret the results. The second is the **`injection-site`**, which captures the details of where the tracer injection was made and is a child of the **neuroanatomical-location** entity. The third entity in question pertains to the location and description of transported labelling (**labeled-location** entities) which include both the location of the label and salient characteristics, such as **density** and **type** ('cellular', 'fiber', 'varicose', *etc.*).

The <<DV>> stereotypes for **labeled-location** class and the **density** and **type** attributes denote that they are considered dependent variables (and may be processed accordingly when required within our knowledge modeling process). The structure of other entities, such as the **neuroanatomical-location** involves potentially several named structures from an atlas (**atlas-volume** objects) since a single location or site may conceivably involve multiple named structures. The **abbr** attribute denotes the abbreviation commonly used to describe the **atlas-volume**, and the **addr** attribute denotes the 'address' of the region (a construct used to denote the position of the region in the hierarchy).

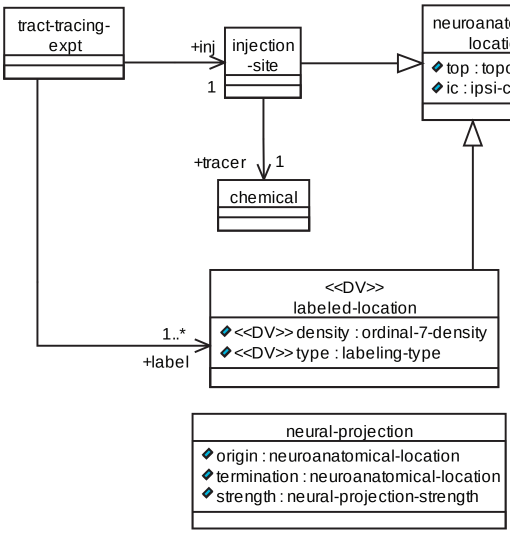


Fig. 2.5. Schema for tract-tracing studies used in the text-mining examples described in this chapter.

In Figure 2.5, note the presence of the **neural-projection** which represents an interpreted 'fact' that takes the form location A projects to location B with strength C. A large number of these facts could be summarized into a large connection matrix and then analyzed mathematically in a form of a model (see Figures 2.2 and 2.3, [78]). The logic required to construct these facts is simple: if the **chemical** is a retrograde tracer, then construct a **neural-projection** originating from the **labeled-location** and terminating in the **injection-site** and vice-versa if the **chemical** is anterograde. This experiment-type therefore provides a suitably simple example for investigating our method's feasibility.

UML is not considered particularly effective as knowledge representation. It is a widely-used inudstrial standard for software engineering and provides an effective way of expressing class/attribute/role relationships diagrammatically. We automatically converted the above schema to OWL format and show the equivalent schema rendered for viewing within the OntoViz plugin of the Protégé ontology editing tool (see Figure 2.6). Using UML presents a low 'barrier to entry' for non-specialists who are familiar with concepts from object-oriented programming.

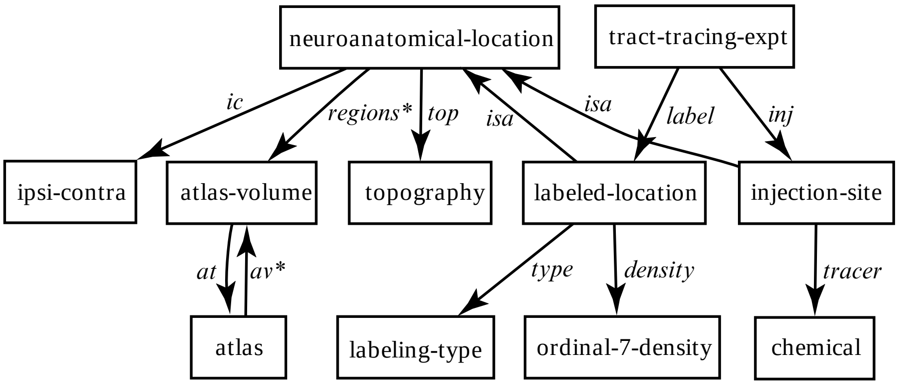
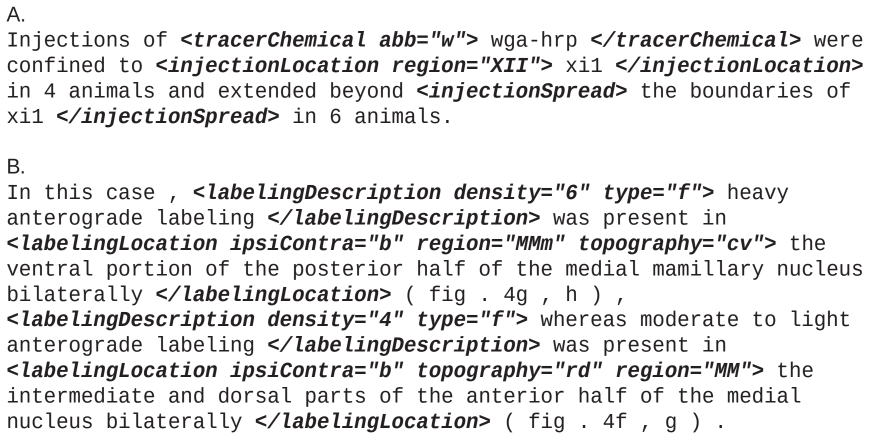


Fig. 2.6. Translation of the UML schema into an OWL ontology, (rendered by OntoViz within Protégé/OWL).

The main challenge is to populate this representation from the textual narrative of published papers. Naturally, the textual descriptions of this data are typically far more complex than our simple schema, involving a wide variety of syntactic and semantic structures (some of which act as modifiers for tags that demarcate the main items). We use an XML-based representation to provide a set of markup tags that capture recognizable linguistic features of entities from our target representation and of additional entities to capture additional structure from the text. In Figure 2.7, we present two examples of text with accompanying XML markup to illustrate our methodology, the first relatively simple, the second more complex and more representative of our input data.

This illustrates how we use XML tags to demarcate phrases of interest (including names of brain structures, descriptions of labeling patterns, topographic terminology, *etc.*). Note that we have invented our new tags to provide additional processing structure for subsequent evaluation. For example, the **`<injectionSpread>`** tag denotes regions that may (or may not) be involved in the injection site. Constructing the tagset is an iterative ongoing process where we initially created a simple representation and then refined it as we mark up the text and run our NLP methods. The XML scheme serves two purposes: to capture the semantic detail sufficiently well to be able to populate the target representation and also to maximize performance of our system's ability to mark up the text. Consequently, there is not a one-to-one



```
A.
Injections of <tracerChemical abb="w"> wga-hrp </tracerChemical> were
confined to <injectionLocation region="XII"> xi1 </injectionLocation>
in 4 animals and extended beyond <injectionSpread> the boundaries of
xi1 </injectionSpread> in 6 animals.

B.
In this case , <labelingDescription density="6" type="f"> heavy
anterograde labeling </labelingDescription> was present in
<labelingLocation ipsiContra="b" region="MMm" topography="cv"> the
ventral portion of the posterior half of the medial mamillary nucleus
bilaterally </labelingLocation> ( fig . 4g , h ) ,
<labelingDescription density="4" type="f"> whereas moderate to light
anterograde labeling </labelingDescription> was present in
<labelingLocation ipsiContra="b" topography="rd" region="MM"> the
intermediate and dorsal parts of the anterior half of the medial
nucleus bilaterally </labelingLocation> ( fig . 4f , g ) .
```

Fig. 2.7. Text from tract-tracing experiments, marked up with XML tags under the current design of the text-mining process. Note the presence of a common 'OCR error' in the text extracted captured from the PDF file in (A): here 'xi1' is really 'XII', the hypoglossal nucleus in the rat.

correspondence between our markup tags and output representation shown in Figure 2.5. Traversing this step involves ongoing research that is currently at a preliminary stage. For the rest of this chapter, we describe our approach to automatically insert these tags into text across a corpus of papers describing tract-tracing experiments using modern NLP techniques.

2.7 Methodology

The overall challenge, from text to knowledge

The text-mining process when applied to our problem of tract-tracing experiments may be decomposed into subtasks: (a) identifying documents of interest, (b) delineating individual experiments in the text, (c) accurately tagging the appropriate text within the narrative, (d) annotating the marked-up text accurately, (e) composing the complete annotation into computable entities conforming to our target schema. Each one of these tasks requires the use of different sets of tools to accomplish the end goal of constructing database entries from the raw text input. Bringing all of these components together is an ongoing task within the project.

Existing computational approaches all contribute to address subtasks of this process. Fortunately, these separate subtasks are being addressed by research performed on several different topics within the community: Document Classification, Named Entity Recognition, Relation Extraction and Event Characterization (following the MUC competitions [87] and more recent competitions within the bio-NLP community [88]). Note that by our existing definitions, the overall task of extracting information pertaining to a specific complete experiment is synonymous with that of Event Detection and Characterization, a task that has been notoriously difficult to solve with high performance in the MUC and ACE evaluations (best performance tends to hover around 60% precision and recall, [89]). Even so, given the importance of biocuration and the high cost of maintaining biomedical databases, developing methods to improve efficency of constructing knowledge bases will still have a large impact.

The overall task has too many subcomponents to describe in detail here. Instead we will focus on the process of automatically inserting appropriate tags into the text and describe the methodology for this in detail.

Information extraction techniques: from Patterns to Conditional Random Fields

For the NLP community, IE has been a constantly active area since the 1970s. IE processes text corpora to populate a target representation, typically recorded in a database. Ideally, the specific information extracted should be concise and may contain several words or a phrase. Much of the current work in IE is pattern-based, that is, specific textual patterns are defined and associated with the data/knowledge types of interest. When the patterns match a fragment of text, they serve both to delimit the region of interest and to allocate it to a specific data type. For example, a pattern can be '<person> was born in <place> on <date>'. Whenever this pattern encounters a matching sentence, the person's name, the birth place and the birth date are extracted. This is the approach used within the Textpresso system [7].

Acquiring meaningful patterns is the key to this approach, and is the main restriction in terms of its usefulness. It is usually hard to create a complete pattern list with all variations that are naturally encountered in human language. Traditionally, these patterns were constructed either manually or programmatically, or they were acquired from human-annotated corpora, *e.g.*, [90–92]. In these cases, it is not generally guaranteed that all possible patterns can be included within such a manually compiled list. Therefore, these approaches tend to have unpredictable coverage. The cost of human annotation in these cases is non-trivial and must be repeated in each domain. It is also the case that required fields do not always follow fixed patterns and patterns cannot be derived with sparse data.

Depending on the extraction task, identifying the required information from the text may require additional knowledge beyond that expressible in a surface text pattern. This limited ability of expressivity arises since the only information represented is a sequence of words in a fixed order. Although some research reported to derive more complex patterns by mixing Part of Speech (POS) tags and surface words [63, 93], patterns cannot be integrated with other types of useful required knowledge, (such as the root form of a word).

A straightforward way to construct patterns is to annotate manually a large number of sentences with the required slots (a laborious task). It is possible to learn surface patterns by bootstrapping from a set of seed data [94, 95]. However, the power of this approach is somewhat limited and at most, only two slots are allowed in a single sentence. The ability to learn patterns with multiple slots has not been yet reported with reasonable performance.

In this bootstrapping procedure, a set of seed data pairs are prepared in advance. For example, we first manually create a list of person names and their birthdates for the relation ‘birth date’. The system then scans a text corpus (or a corpus returned by querying search engines with these terms). Any sentence containing both search terms is automatically identified. Slots are renamed with two anchor names, for example, <person> and <birthdate> respectively in this case. A suffix tree traverse algorithm [96] then builds a suffix tree to strip off non-common portions of the sentences, leaving potential patterns. The learned patterns can be verified with a small validation set and used to extract relations in the future. Systems requiring averagely ten to twenty seed data pairs can obtain promising results while significantly reducing expensive human costs.

As mentioned above, pattern-based approaches learn useful patterns to pinpoint required fields values using seed data. Most approaches on binary relation extraction [94, 97] rely on the co-occurrence of two recognized terms as anchors in a single sentence. However, this approach cannot be generalized to more complex situations where the data corpus is not rich enough to learn variant surface patterns. Although surface pattern based techniques perform well when sentences are short and complete pattern lists are available, sentences within biomedical articles tend to be long and the prose structure tends to be complex, reducing the effectiveness of short contiguous patterns. What is required is the ability to recognize automatically, sequences of important indicator terms, regardless of intermediate material and then to assemble the pertinent parts as required.

A promising development within NLP research is the Conditional Random Field (CRF) model for sequential labeling [97, 98], which has been widely used for language processing, including improved model variants [99], web data extraction [100], scientific citation extraction [101], and word alignment [102]. The originators of the CRF model provide an open-source implementation in the form of the MALLET toolkit [103].

As given in [104], this model is simply a conditional probability $P(y|x)$ with an associated state-based graph structure. Within a labeling task, where the model traverses a set of tokens (words) and labels each one according to the current state of the model, the most likely transitions between states (to given the most likely label assigned to a given token) is given by summing a set of weighted feature functions. Here each feature, defined by the system builder, reflects some (potentially) pertinent aspect of the text: it may be a word, a part of speech, semantic or syntactic label, punctuation mark, formatting command, *etc.*

The structure of this graph is adaptable and when it takes the form of a linear chain, the CRF model has very similar properties to Hidden Markov Models (HMMs). CRF models have inherent characteristics that outperform the limitations of old pattern based approaches: they view all the required knowledge to extract useful information as features and given reasonable training data, they compile those features automatically to extract information. They can provide a compact way to integrate many different types of features (including explicit surface word patterns).

Therefore, CRF models have more powerful expressivity even when potential patterns have never been seen before by the system. In this situation, the CRF models utilize related information from heterogeneous sources. Additionally, they do not suffer from any limitation to the number of slots per sentence.

We use plain text as a token sequence for input and attempt to label each token with field labels. For each current state, we train the conditional probability of its output states given previously assigned values of input states. Formally, given a sentence of a separate input sequence of word tokens, $S = (w_1, w_2, ..., w_n)$, we attempt to obtain a corresponding labeling sequence of field names, $L = (l_1, l_2, ..., l_n)$, and each input token corresponds to only one label. The field names must include a default label, ‘O’, denoting that the token receives a null label.

The CRF model is trained to find the most probable labeling sequence L for an input sentence S by maximizing the probability of $P(L|S)$. The decision rule for this procedure is:

$$\hat{L} = \arg\max_{L} P(L|S) \tag{2.1}$$

As described above, this CRF model is characterized by a set of feature functions and their corresponding weights. The conditional probability can be computed using Equation 2.2 (as with Markov fields).

$$P(L|S) = \frac{1}{Z_S} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k * f_k(l_{t-1}, l_t, S, t)\right) \tag{2.2}$$

Where $f_k(l_{t-1}, l_t, S, t)$ is a feature function, including both the state transition feature $f_k(l_{t-1}, l_t, S)$ and the feature of output state given the input sequence $f_k(l_t, S)$. A detailed introduction to the mathematics of this formulation may be found in [98, 104] and will not be repeated here.

The individual feature functions are created by the system builder, often using standard computational linguistic tools such as parts of speech (POS) taggers, parsers, lexions *etc.* Since the CRF model automatically learns which features are relevant for which labels, the system builder is free to experiment with a variety of features.

The methodology we use is based on a supervised-learning approach. In order to learn which features predict which label(s), the CRF model requires a prelabeled training set to learn and optimize system parameters. To avoid the over-fitting problem, a Gaussian prior over the parameters is typically applied to penalize the log-likelihood [105]. We calculate the first-derivative of this adjusted log-likelihood value, and use it to maximize this probability and estimate the values for λ_k. Once we have obtained these parameter values, the trained CRF model can be used to make predictions with previously unseen text.

The principal obstacle to the development of general-purpose information extraction systems based on supervised approaches is that obtaining enough suitably formatted training data upon which to train the system is either too expensive or too complex. Besides the bootstrapping approaches described previously, the procedure called 'active learning' strives to reduce the annotation cost by selecting the most informative training examples and presenting just them to the annotators, thereby obtaining from them the maximally useful feedback. Once these informative examples have been constructed they are added to the training set to improve system performance. Thus, one can start with a small annotation set, train a system, provide annotators with initial, largely inaccurate system results, and then use the corrections provided by the annotators to refine the learning process. Active learning approaches may be categorized on the basis of the selection criteria concerning data to be cycled through the annotation/correction process. 'Committee-based' approaches select data with the highest disagreement (in the classification task) to be reprocessed. 'Uncertainty/certainty score-based' approaches, require that all new data undergoing classification are assigned uncertainty/certainty scores based on predefined measurements. Those data with the highest uncertainty scores are then returned to be reprocessed by a human annotator and added to the training set.

These methods can significantly accelerate the process of annotating training data for machine learning systems. Although these methods were initially introduced into language processing for classification tasks [106, 107] many different NLP fields have adopted this idea to reduce the cost of training. These include information extraction and semantic parsing [108]; statistical parsing [109]; Named Entity Recognition [110]; and Word Sense Disambiguation [111].

Stage 1: Corpus Processing

The first stage of performing text mining work is to obtain and preprocess as large a body of textual information as possible. This involves downloading research articles from the web and extracting the text of these articles to remove the formatting used by individual journals. This is a vital but non-trivial step since the articles may only be available in formats that are difficult to manipulate (such as PDF). In addition to this, the publishing process places breaks, figure- and table-legends into the flow of the text so that an uninterrupted stream of the experimental narrative is not directly readable from the file without additional processing. Within our example application concerned with neuroanatomical connections, we used a geometric, rule-based approach built on top of a well-engineered, open-source document management system ('Multivalent' [112]) to parse the PDF files that make up our corpus.

We use the Journal of Comparative Neurology as the basis for our text corpus. This is an authoritative publication for neuroanatomists and for neuroscience in general [113]. We acted within the journal's copyright guidelines to download roughly 12,000 articles dating from 1970 to 2005. This coincides with the timeframe over which tract-tracing experiments have been performed. We used papers that had consistent formatting from volume 204 to 490 (1982-2005) providing a complete text corpus of 9,474 files and 99,094,318 words distributed into various article sections ('Abstract', 'Introduction', *etc.*). We restricted our attention to the results sections of these papers, which comprised roughly one quarter of the total text in the corpus.

As with many other researchers in the field of biomedical text mining, the preferred representation for text data is the Extensible Markup Language (XML). There are preexisting XML editors that support the process of adding annotations to text (see, for example, the Vex system [114]). This provides a convenient standardized user interface for the process of annotation (which is the most time-consuming and tedious component of this type of work). Vex also uses standard web-formatting (Cascading Style Sheets or 'CSS') to permit the user to define their own visual formatting for the text being annotated. For a review of tools to assist with managing corpora and their annotation, see [115].

Stage 2: The basic text processing and feature definition

In order to locate the text that pertains to the semantic structures specified in the schema, we use a set of morphological, lexical, grammatical or semantic functions to provide features that can train the CRF model to tag words appropriately. These feature functions implement NLP-based functions and may use downloadable tools within their implementation. The binary functions that we define for our target application are as follows:

- **Lexical features:** Entities defined within the schemas affiliated with each type of experiment are often identifiable through specific names. Note that in many domains that have not been centralized and regulated, the nomenclature of concepts is often quite messy, exceedingly complicated and contradictory (see [116, 117] for examples from neuroanatomy and [118, 119] for a discussions of this topic in the context of Named Entity Recognition for molecular biology). For our work with tract-tracing experiments, we use lexicons that are pre-chosen for different components of the schema. These include named neuroanatomical structures (taken from [120]), the neuroanatomical cardinal directions[1] (*e.g.*, 'caudal', 'rostral', *etc.*), names and abbreviations of tracer chemicals used (*e.g.*, 'PHAL'), and commonsense words that describe the density of labeling (*e.g.*, 'dense', 'weak', *etc.*). Given a different schema, we would select different features and would construct lexica from the expanding number of biomedical controlled vocabularies that are now avaiable (see [121] for review). Every word in a given lexicon forms a separate feature (*e.g.*, the feature **`lexicon-region`** only returns 1 if the word being labeled appears in the 'region' lexicon).
- **Surface and window words:** We employ the word itself and the immediately surrounding words as features for the labeling algorithm, (*e.g.*, the feature **`surface-injection`** only returns 1 if the word being labeled is 'injection'; the feature **`previous-injection`**, returns 1 only if the previous word is 'injection'; the **`next-injection`** feature function acts similarly for the following word).

[1] Six terms are typically used to denote directions along the three orthogonal axes within neuroanatomy: 'rostral' and 'caudal' denote the front-to-back direction; 'medial' and 'lateral' denotes towards or away from the midline; and 'dorsal' and 'ventral' denotes the top-to-bottom direction. These terms are often used in combination, so that 'dorsolateral' refers to a direction to the top and away from the midline.

- **Dependency relations:** We use a dependency parser ('MiniPar' [122]) to parse each sentence, and then derive four types of features from the parsing result. The first type is the root forms of words when this differs from the presented form (*e.g.*, the feature **`root-inject`**, only returns 1 if the being labeled is 'injected', 'injection' or some other derivation and 0 otherwise). The second and third types of dependency feature are based on the subject and object of the sentence. For example, the feature **`syntax-subject`**, only returns 1 if the word being labeled is the subject of the sentence. Similarly, the feature **`syntax-object`** only returns 1 if the word is the object of the phrase. The fourth type of feature is based on the governing verb for each word. We traverse dependency relations to the nearest verb within the parse and then base the definition of our feature on the root form of that verb (*e.g.*, the feature **`govern-inject`**, only returns 1 if the word being labeled is governed by the verb 'to inject' in the target sentence).

It should be remembered that the choice of features is not restricted for the set described above. These simply provide a working model for our tract-tracing example. Other types of features that can be used may be based on parts of speech, character n-grams, word-shapes, previously tagged words, short forms of abbreviations and other variants (see [123] as an example with a widespread choice of features). The choice of feature functions largely determines the performance of the system and is where the system designer can most greatly influence the outcome of the text-mining process. It is also possible to view the weight parameters for each individual feature for each state transition within the CRF model. This provides possible powerful insight into the reasoning learned within the labeling task.

Interestingly, studies of automated methods that evaluate the reliability of curated facts within the Geneways system also consider relationships and clustering between features themselves [124]. This illustrates the versatility and power of machine learning approaches in this context. Biomedical experts contribute to this process by providing training data for which computer scientists and NLP-experts may then devise suitable features.

Stage 3: Evaluating the results

As is usually the case within the field of NLP, quantitative evaluation is essential for the development of the techniques described here. The standard measurements that are most relevant to this question are measures of inter-annotator agreement (often based on the kappa statistic, [125]). This is calculated in the following formula where P(A) is equal to the proportion of times annotators agree, and P(E) is equal to the proportion of times annotators would be expected to agree according to chance alone.

$$Kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.3}$$

The recall (what proportion of target items have been correctly extracted?) and the precision (how many of these extracted items were themselves correct?) are routinely measured and reported for IE tasks. These two values may also be expressed as a single number by the use of the F-Score, see [126, 127].

$$Precision = \frac{\text{\# of correct extracted items}}{\text{\#of all extracted items}} \tag{2.4}$$

$$Recall = \frac{\text{\# of correct extracted items}}{\text{\#of target items from the `gold standard' reference}} \tag{2.5}$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.6}$$

Methods of evaluation within the field as a whole centers around shared evaluations where the task being addressed is standardized and presented to the community as a competition. Within biomedicine, recent evaluations include the KDD challenge cup task 1 (2002) [69], the TREC genomics track (2003-2006) [70, 71], BioCreAtIvE (2003-2004, 2006) [128].

Extrinsic measures of evaluation provide feedback from the perspective of domain experts and are based on subjective criteria such as ‘is this useful?’, and ‘does this save time?’. Evaluating these type of criteria must depend on subject interviews and questionnaires. It is also possible to record behavioral statistics from the use of the system, which can provide valuable data to indicate how well the system performs at a specific task.

We focus on the development of systems that must be developed, implemented and tested. Whilst engaged in this pursuit, we use four extrinsic evaluation tasks. (1) requirements evaluation (are requirements fully specified, complete and attainable?); (2) system validation and verification (does the system fully represent the knowledge it is supposed to, and is the system built well technically?); (3) usability evaluation (is the system easy to learn and use?); (4) performance evaluation (how well does the system fulfill its requirements?). One important measure that we emphasize is the time taken to annotate documents by domain experts. Usage metrics (such as the number of system downloads over a given period) can also provide insight as to the impact of a given system on the community [129, 130].

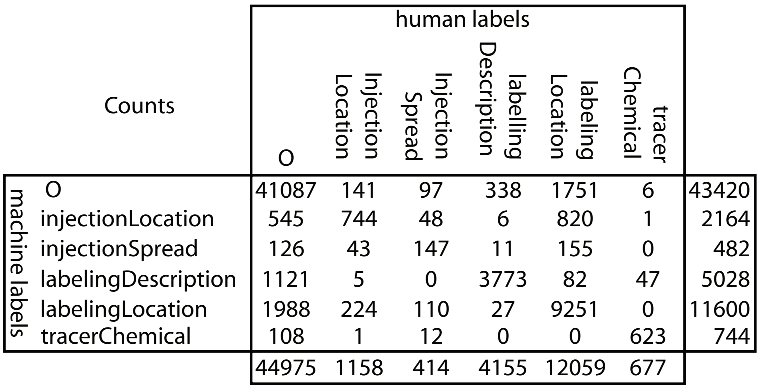
2.8 Results

In a series of experiments, we marked up 21 documents by hand, providing 1047 sentences (at an approximate rate of 45 sentences per hour). We then randomly divided this data into training and testing sets (with 698 and 349 sentences respectively) to reconstruct our annotations. The system’s performance based on different combinations of features is shown in Table 2.1. Performance of this task is acceptably high (F-Score = 0.79). This is especially encouraging because the number of training examples (14 documents) is relatively small. We then ran our labeling system on previously unseen text and corrected the machine driven annotations by hand. We found that this process had been accelerated to an approximate rate of 115 sentences per hour.

Table 2.1. NLP performance (Precision, Recall and F-Score) for text-mining from tract-tracing experiments. Features Key, L = Lexical, C = Current Word, P/F = Preceding or Following Word, W = Context Window, D = Dependency features.

Features	Precision	Recall	F-Score
Base	0.41	0.18	0.25
L	0.60	0.37	0.46
L + C	0.77	0.73	0.75
L + C + P/F	0.77	0.73	0.75
L + C + P/F + W	0.81	0.75	0.78
L + C + P/F + W + D	0.80	0.78	0.79

The confusion matrix describing the errors made by our system is shown in Figure 2.8. The leading diagonal holds the counts of our system's correct guesses for word-labels, and off-diagonal counts demonstrate errors. Note that the three labels for different types of neuroanatomical locations are frequently confused (**`<injectionLocation>`**, **`<tracerChemical>`**, and **`<labelingLocation>`**). Pooling these labels into a single category yields Recall = 0.81, Precision = 0.85, F-Score = 0.83. This emergent property of the textual descriptions may also have relevance to the design of the knowledge representations and ontological resources derived from text. Annotating the text of articles involves developing suitable labeling schemes and performing a large amount of annotation. This is particularly instructive concerning subtleties of representation that may be relevant to ontology engineers. We envisage that much of our future work will center on the actions of biocurators as knowledge engineers, enabled in their work by NLP approaches and KB systems.



		human labels						
Counts		O	Injection Location	Injection Spread	labelling Description	labeling Location	tracer Chemical	
machine labels	O	41087	141	97	338	1751	6	43420
	injectionLocation	545	744	48	6	820	1	2164
	injectionSpread	126	43	147	11	155	0	482
	labelingDescription	1121	5	0	3773	82	47	5028
	labelingLocation	1988	224	110	27	9251	0	11600
	tracerChemical	108	1	12	0	0	623	744
		44975	1158	414	4155	12059	677	

Fig. 2.8. A 'confusion matrix' for the tract-tracing experimental data.

2.9 Conclusions

This chapter is concerned with ‘Intelligent Text Mining’; thus, the main component of our work described here is to describe an appropriate target for our text mining approaches. The central concept of this work is the view shown in Figure 2.2: we base our methodology primarily on experimental observations (that may be used to construct representations of ‘facts’ and ‘models’). Each individual **experiment**, consists of a set of **independent variables** (that capture the constraints imposed on the experiment) and a set of **dependent variables** (that capture the measurements made within the experiment). Commonly used experimental designs provide templates for specific experiment-types that can be used to create effective data summaries of experimental observations.

Despite the astonishing scholarly abilities of top-level biologists, the number of individual experimental facts that typically pertain to any phenomenon of interest taxes the limits of human memory. The overall objective of this work is to provide large-scale knowledge-bases that serve as massive literature reviews. We observe that the process of constructing such systems mimics the process of performing a meta-analysis of the literature for specific experimental-types. Once such a knowledge-base has been compiled, new, previously unseen summaries of research data provide insight that is only possible from data-mining of such large-scale systems (see [3] for mathematical meta-analyses of neural connectivity based on these summary data).

The rate-determining step facing workers building such systems is knowledge acquisition, and many existing biomedical databases rely solely on the highly expensive process of human curation. This provides the underlying need for text-mining tools that fit can supply appropriately structured data. An interesting challenge of building these text-mining approaches lies in the possibility of providing tools that can be used by biocurators, which may then leverage their expertise and dedication into their functionality. It is crucial that advances in computer science translate effectively into application development for academic biomedical informatics systems. In our example of neuroanatomical tract-tracing experiments, we provide a system that may be used to support specific databases for this experimental-type [4, 5]. Given the performance of our system (F-Score = 0.79 for the text labeling task), we would not expect to provide a completely automated solution, but a significant increase in biocuration efficiency may permit these system-developers to provide a more comprehensive account of the literature with fewer curation resources.

It is currently an exciting time in the field of biomedical knowledge engineering. Advances in the performance and versatility of open-source machine-learning systems [98, 103], and in the maturity and infrastructure surrounding the use of ontologies in biomedicine [15] provide a rich, highly collaborative and productive environment for future work. We hope that this chapter encourages computer scientists to address these important questions through the development of new approaches and tools.

2.10 Acknowledgements

This work was funded under grant LM 07061 from the NIH and from seed funding generously provided by the Information Sciences Institute. We are grateful for the support and vision of the executive leadership of the Institute. The central role of independent and dependent variables within our strategy is attributed to discussion with Prof. Alan Watts at USC. The central theme of knowledge statements within our knowledge representations was arrived at in brainstorming sessions with Dr Arshad Khan. Many of the principles underlying this project emerged from lengthy discussions with the Watts and Swanson laboratories at USC. We are also grateful to Patrick Pantel for his good-natured brilliance and general support with this project.

References

[1] J. P. Jenuth (2000), Methods Mol Biol, 132: 301-12

[2] I. Sim, G. D. Sanders and K. M. McDonald (2002), J Gen Intern Med, 17(4): 302-8

[3] M. P. Young, J. W. Scannell and G. A. P. C. Burns (1995), The analysis of cortical connectivity. ed. ed. Vol., Austin, Texas: R. G. Landes.

[4] K. E. Stephan, L. Kamper, A. Bozkurt, G. A. Burns, M. P. Young and R. Kotter (2001), Philos Trans R Soc Lond B Biol Sci, 356(1412): 1159-86

[5] M. Bota, H. Dong and L. W. Swanson (2005), Neuroinformatics, 3(1): 15-48

[6] T. Reguly, A. Breitkreutz, L. Boucher, B. J. Breitkreutz, G. C. Hon, C. L. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong, C. Stark, Y. Ho, D. Botstein, B. Andrews, C. Boone, O. G. Troyanskya, T. Ideker, K. Dolinski, N. N. Batada and M. Tyers (2006), J Biol, 5(4): 11

[7] H. M. Muller, E. E. Kenny and P. W. Sternberg (2004), PLoS Biol, 2(11): e309

[8] A. Rzhetsky, I. Iossifov, T. Koike, M. Krauthammer, P. Kra, M. Morris, H. Yu, P. A. Duboue, W. Weng, W. J. Wilbur, V. Hatzivassiloglou and C. Friedman (2004), J Biomed Inform, 37(1): 43-53

[9] R. Kotter (2004), Neuroinformatics, 2(2): 127-44

[10] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi and L. S. Yeh (2004), Nucleic Acids Res, 32(Database issue): D115-9

[11] M. Kanehisa (2002), Novartis Found Symp, 247: 91-101; discussion 101-3, 119-28, 244-52

[12] H. Cooper and L. V. Hedges (1994), The Handbook of Research Synthesis. ed. ed. Vol., New York: Russell Sage Foundation.

[13] G. A. Burns, A. M. Khan, S. Ghandeharizadeh, M. A. O'Neill and Y. S. Chen (2003), Neuroinformatics, 1(1): 81-109

[14] G. A. Burns and W. C. Cheng (2006), J Biomed Discov Collab, 1(1): 10

[15] D. L. Rubin, S. E. Lewis, C. J. Mungall, S. Misra, M. Westerfield, M. Ashburner, I. Sim, C. G. Chute, H. Solbrig, M. A. Storey, B. Smith, J. Day-Richter, N. F. Noy and M. A. Musen (2006), Omics, 10(2): 185-98

[16] T. R. Gruber (1993), Towards principles for the design of ontologies used for knowledge sharing. in International Workshop on Formal Ontology. Padova, Italy.

[17] N. F. Noy, M. Crubezy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti and M. A. Musen (2003), AMIA Annu Symp Proc: 953

[18] D. Oberle, R. Volz, B. Motik and S. Staab (2004), An extensible ontology software environment, In, Handbook on Ontologies, Springer. 311-333.

[19] OBO-Edit - The OBO Ontology Editor [http://oboedit.org/]

[20] D. R. Swanson (1990), Bull Med Libr Assoc, 78(1): 29-37

[21] M. V. Blagosklonny and A. B. Pardee (2002), Nature, 416(6879): 373

[22] N. R. Smalheiser and D. R. Swanson (1998), Comput Methods Programs Biomed, 57(3): 149-53

[23] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin and G. Sherlock (2000), Nat Genet, 25(1): 25-9

[24] B. Latour and S. Woolgar (1979), Laboratory Life. 2 ed. ed. Vol., Princeton, New Jersey: Princeton University Press.

[25] G. D. Ruxton and N. Colegrave (2003), Experimental design for the life sciences. ed. ed. Vol., Oxford: Oxford University Press.

[26] S. S. Stevens (1946), Science, 103(2684): 677-680

[27] D. A. Sprott (2000), Statistical Inference in Science. ed. ed. Vol., New York: Springer Verlag.

[28] N. F. Noy, M. Crubezy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti and M. A. Musen (2003), AMIA Annu Symp Proc: 953

[29] P. D. Karp (2001), Science, 293(5537): 2040-4

[30] C. Brooksbank and J. Quackenbush (2006), Omics, 10(2): 94-9

[31] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo and M. Vingron (2001), Nat Genet, 29(4): 365-71

[32] E. W. Deutsch, C. A. Ball, G. S. Bova, A. Brazma, R. E. Bumgarner, D. Campbell, H. C. Causton, J. Christiansen, D. Davidson, L. J. Eichner, Y. A. Goo, S. Grimmond, T. Henrich, M. H. Johnson, M. Korb, J. C. Mills, A. Oudes, H. E. Parkinson, L. E. Pascal, J. Quackenbush, M. Ramialison, M. Ringwald, S. A. Sansone, G. Sherlock, C. J. Stoeckert, Jr., J. Swedlow, R. C. Taylor, L. Walashek, Y. Zhou, A. Y. Liu and L. D. True (2006), Omics, 10(2): 205-8

[33] J. Leebens-Mack, T. Vision, E. Brenner, J. E. Bowers, S. Cannon, M. J. Clement, C. W. Cunningham, C. dePamphilis, R. deSalle, J. J. Doyle, J. A. Eisen, X. Gu, J. Harshman, R. K. Jansen, E. A. Kellogg, E. V. Koonin,

B. D. Mishler, H. Philippe, J. C. Pires, Y. L. Qiu, S. Y. Rhee, K. Sjolander, D. E. Soltis, P. S. Soltis, D. W. Stevenson, K. Wall, T. Warnow and C. Zmasek (2006), Omics, 10(2): 231-7

[34] G. A. Burns (2001), Philos Trans R Soc Lond B Biol Sci, 356(1412): 1187-208

[35] R. Homayouni, K. Heinrich, L. Wei and M. W. Berry (2005), Bioinformatics, 21(1): 104-15

[36] R. B. Norgren, Jr. and M. N. Lehman (1998), Neurosci Biobehav Rev, 22(6): 695-708

[37] M. Y. Galperin (2006), Nucleic Acids Res, 34(Database issue): D3-5

[38] The Neuroscience Database Gateway [http://ndg.sfn.org/]

[39] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin and N. Lopez-Bigas (2005), Nucleic Acids Res, 33(19): 6083-9

[40] R. Caspi, H. Foerster, C. A. Fulcher, R. Hopkinson, J. Ingraham, P. Kaipa, M. Krummenacker, S. Paley, J. Pick, S. Y. Rhee, C. Tissier, P. Zhang and P. D. Karp (2006), Nucleic Acids Res, 34(Database issue): D511-6

[41] I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil and P. D. Karp (2005), Nucleic Acids Res, 33(Database issue): D334-7

[42] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz and M. Tyers (2006), Nucleic Acids Res, 34(Database issue): D535-9

[43] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono and M. Kanehisa (1999), Nucleic Acids Res, 27(1): 29-34

[44] L. Kreppel, P. Fey, P. Gaudet, E. Just, W. A. Kibbe, R. L. Chisholm and A. R. Kimmel (2004), Nucleic Acids Res, 32(Database issue): D332-3

[45] M. Ashburner and R. Drysdale (1994), Development, 120(7): 2077-9

[46] J. A. Blake, J. E. Richardson, M. T. Davisson and J. T. Eppig (1997), Nucleic Acids Res, 25(1): 85-91

[47] S. Twigger, J. Lu, M. Shimoyama, D. Chen, D. Pasko, H. Long, J. Ginster, C. F. Chen, R. Nigam, A. Kwitek, J. Eppig, L. Maltais, D. Maglott, G. Schuler, H. Jacob and P. J. Tonellato (2002), Nucleic Acids Res, 30(1): 125-8

[48] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng and D. Botstein (1998), Nucleic Acids Res, 26(1): 73-9

[49] E. Huala, A. W. Dickerman, M. Garcia-Hernandez, D. Weems, L. Reiser, F. LaFond, D. Hanley, D. Kiphart, M. Zhuang, W. Huang, L. A. Mueller, D. Bhattacharyya, D. Bhaya, B. W. Sobral, W. Beavis, D. W. Meinke, C. D. Town, C. Somerville and S. Y. Rhee (2001), Nucleic Acids Res, 29(1): 102-5

[50] E. F. Kirkness and A. R. Kerlavage (1997), Methods Mol Biol, 69: 261-8

[51] T. W. Harris, R. Lee, E. Schwarz, K. Bradnam, D. Lawson, W. Chen, D. Blasier, E. Kenny, F. Cunningham, R. Kishore, J. Chan, H. M. Muller, A. Petcherski, G. Thorisson, A. Day, T. Bieri, A. Rogers, C. K. Chen, J. Spieth, P. Sternberg, R. Durbin and L. D. Stein (2003), Nucleic Acids Res, 31(1): 133-7

[52] M. Westerfield, E. Doerry, A. E. Kirkpatrick and S. A. Douglas (1999), Methods Cell Biol, 60: 339-55

[53] P. E. Bourne and J. McEntyre (2006), PLoS Comput Biol, 2(10): e142

[54] The Jackson Laboratory - Advancing Research in Human Health [http://www.jax.org/]

[55] J. D. Kim, T. Ohta, Y. Tateisi and J. Tsujii (2003), Bioinformatics, 19 Suppl 1: i180-2

[56] A. Yakushiji, Y. Tateisi, Y. Miyao and J. Tsujii (2001), Pac Symp Biocomput: 408-19

[57] P. D. Karp (2000), Bioinformatics, 16(3): 269-85

[58] The Signal Transduction Knowledge Environment (STKE) [http://stke.sciencemag.org/]

[59] A. Rzhetsky, T. Koike, S. Kalachikov, S. M. Gomez, M. Krauthammer, S. H. Kaplan, P. Kra, J. J. Russo and C. Friedman (2000), Bioinformatics, 16(12): 1120-8

[60] C. Friedman, P. Kra, H. Yu, M. Krauthammer and A. Rzhetsky (2001), Bioinformatics, 17 Suppl 1: S74-82

[61] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, L. Y. Geer, W. Helmberg, Y. Kapustin, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, K. D. Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner and E. Yaschenko (2006), Nucleic Acids Res, 34(Database issue): D173-80

[62] A. Gass (2004), PLoS Biol, 2(10): e353

[63] P. Pantel, D. Ravichandran and E. H. Hovy (2004), Towards Terascale Knowledge Acquisition. in Proceedings of the COLING conference. Geneva, Switzerland.

[64] A. Khan, J. Hahn, W.-C. Cheng, A. Watts and G. Burns (2006), Neuroinformatics, 4(2): 139-160

[65] W.-C. Cheng and G. A. P. C. Burns (2006), NeuARt II Developers Manual, In, University of Southern California, Information Sciences Institute.: Los Angeles. 1-44.

[66] H. Cooper (1998), Synthesizing Research. A Guide for Literature Reviews. 3 ed. ed. Vol., Thousand Oaks: Sage Publications.

[67] G. Schrieber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga (2000), Knowledge Engineering and Management. The CommonKADS Methodology. ed. ed. Vol., Cambridge, MA: MIT Press.

[68] D. Rebholz-Schuhmann, H. Kirsch, M. Arregui, S. Gaudan, M. Rynbeek and P. Stoehr (2006), Nat Biotechnol, 24(8): 902-3

[69] A. S. Yeh, L. Hirschman and A. A. Morgan (2003), Bioinformatics, 19 Suppl 1: i331-9

[70] A. M. Cohen and W. R. Hersh (2006), J Biomed Discov Collab, 1: 4

[71] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts and M. Hearst (2005), TREC 2005 Genomics Track Overview. in Text REtrieval Conference (TREC) 2005. Gaithersburg, Maryland.

[72] S. J. Buckingham Shum, V. Uren, G. Li, J. Domingue and E. Motta (2003), Visualizing Internetworked Argumentation, In, Visualizing Argumentation, software tools for collaborative and educational sense making, Springer: London.

[73] T. Chklovski, Y. Gil, V. Ratnakar and J. Lee (2003), TRELLIS: Supporting Decision Making via Argumentation in the Semantic Web. in 2nd International Semantic Web Conference (ISWC 2003). Sanibel Island, Florida, USA. 20-23.

[74] I. Sim, G. D. Sanders and K. M. McDonald (2002), J Gen Intern Med, 17(4): 302-8

[75] P. Radivojac, N. V. Chawla, A. K. Dunker and Z. Obradovic (2004), J Biomed Inform, 37(4): 224-39

[76] W. P. Kuo, E. Y. Kim, J. Trimarchi, T. K. Jenssen, S. A. Vinterbo and L. Ohno-Machado (2004), J Biomed Inform, 37(4): 293-303

[77] R. Durbin, S. Eddy, A. Krogh and G. Mitchison (1998), Biological Sequence Analysis. ed. ed. Vol., Cambridge: University Press, Cambridge.

[78] M. P. Young (1992), Nature, 358(6382): 152-5

[79] G. A. Burns and M. P. Young (2001), Philos Trans R Soc Lond B Biol Sci, 355(1393): 55-70

[80] G. Hirst, C. DiMarco, E. H. Hovy and K. Parsons. (1997), Authoring and Generating Health-Education Documents that are Tailored to the Needs of the Individual Patient. in The Sixth International Conference on User Modeling (UM97). Sardinia, Italy.

[81] W. M. Cowan (1971), Brain Reserach, 37(1): 21-51

[82] C. Kobbert, R. Apps, I. Bechmann, J. L. Lanciego, J. Mey and S. Thanos (2000), Prog Neurobiol, 62(4): 327-51

[83] D. J. Felleman and D. C. Van Essen (1991), Cereb Cortex, 1(1): 1-47

[84] G. A. P. C. Burns (1997), Neural connectivity in the rat: theory, methods and applications, In, Department of Physiological Sciences, Oxford University: Oxford. 1-481.

[85] C. R. Gerfen and P. E. Sawchenko (1984), Brain Res, 290(2): 219-38

[86] L. C. Schmued (1989), Fluoro-Gold and 4-acetamindo-4'-isothiocyanostilbene-2-2'-disulpfonic acid: Use of substituted stilbenes in neuroanatomical studies., In, Methods in Neurosciences, Vol. 3. Quantitative and Qualitative Microscopy., Academic Press: New York.

[87] MUC (1988-95), Overview. in Message Understanding Conference (MUC). http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

[88] L. Hirschman and C. Blaschke (2006), Evaluation of Text Mining in Biology, In, Text Mining for Biology and Biomedicine, Artech House: Boston.

[89] D. Appelt and D. Israel (1999), Introduction to Information Extraction. in IJCAI-99. Stockholm, Sweden. http://www.ai.sri.com/appelt/ie-tutorial/.

[90] S. Soderland, D. Fisher, J. Aseltine and W. Lehnert (1995), CRYSTAL: Inducing a conceptual dictionary. in The Fourteenth International Joint Conference on Artificial Intelligence. Montreal, Canada.

[91] E. Riloff (1993), Automatically constructing a dictionary for information extraction tasks. in The 11th National Conference on Artificial Intelligence. Menlo Park, Calif. 811-816.

[92] J. Kim and D. Moldovan (1993), Acquisition of semantic patterns for information extraction from corpora. in The Ninth IEEE Conference on Artificial Intelligence for Applications.

[93] E. Riloff (1996), Automatically Generating Extraction Patterns from Untagged Text. in The Thirteenth National Conference on Artificial Intelligence (AAAI-96). 1044-1049.

[94] D. Ravichandran and E. Hovy (2002), Learning Surface Text Patterns for a Question Answering System. in Proceedings of the ACL conference. Philadelphia, PA.

[95] D. Feng, D. Ravichandran and E. H. Hovy (2006), Mining and Re-ranking for Answering Biographical Queries on the Web. in Proceedings of the Twenty-First National Conference on Artificial Intelligence.

[96] P. Weiner (1973), Linear Pattern Matching Algorithms. in 14th IEEE Annual Symp. on Switching and Automata Theory. 1-11.

[97] G. S. Mann and D. Yarowsky (2005), Multi-field information extraction and cross-document fusion. in The annual meeting for the Association for Computational Linguistics (ACL-2005). Ann Arbor, MI.

[98] J. Lafferty, A. McCallum and F. Pereira (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. in Proceedings of the International Conference on Machine Learning.

[99] F. Jiao, S. Wang, C. Lee, R. Greiner and D. Schuurmans (2006), Semi-supervised conditional random fields for improved sequence segmentation and labeling. in The annual meeting for the Association for Computational Linguistics (ACL-2006).

[100] D. Pinto, A. McCallum, X. Wei and W. B. Croft (2003), Table Extraction Using Conditional Random Fields. in Proceedings of the ACM SIGIR.

[101] F. Peng and A. McCallum (2004), Accurate information extraction from research papers using conditional random fields. in Proceedings of HLT-NAACL. 329-336.

[102] P. Blunsom and T. Cohn (2006), Discriminative word alignment with conditional random fields. in The annual meeting for the Association for Computational Linguistics (ACL-2006).

[103] Mallet - Advanced Machine Learning for Language [http://mallet.cs.umass.edu/]

[104] C. Sutton and A. McCallum (2006), An Introduction to Conditional Random Fields for Relational Learning., In, In Introduction to Statistical Relational Learning., MIT Press.

[105] S. Chen and R. Rosenfeld (1999), A Gaussian prior for smoothing maximum entropy models, In, Technical Report CMUCS-99-108,, Carnegie Mellon University.

[106] D. D. Lewis and W. A. Gale (1994), A sequential algorithm for training text classifiers. in International Conference on Research and Development in Information Retrieval (SIGIR 1994). Dublin, Ireland.

[107] S. Tong and D. Koller (2000), Support vector machine active learning with applications to text classification. in Seventeenth International Conference on Machine Learning. Stanford University.

[108] C. A. Thompson, M. E. Califf and R. J. Mooney (1999), Active learning for natural language parsing and information extraction. in The Sixteenth International Conference on Machine Learning. Bled, Slovenia.

[109] M. Tang, X. Luo and S. Roukos (2002), Active learning for statistical natural language parsing. in Annual Meeting of the Association for Computational Linguistics (ACL-02). Pennsylvania, PA.

[110] D. Shen, J. Zhang, J. Su, G. Zhou and C. L. Tan (2004), Multi-criteria-based active learning for named entity recognition. in Annual Meeting of the Association for Computational Linguistics (ACL-04). Barcelona.

[111] J. Chen, A. Schein, L. Ungar and M. Palmer (2006), An empirical study of the behavior of active learning for word sense disambiguation. in Human Language Technology conference - North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL) 2006. New York, NY.

[112] T. A. Phelps and R. Wilensky (2001), The Multivalent Browser: A Platform for New Ideas. in Document Engineering 2001. Atlanta, Georgia.

[113] C. B. Saper (1999), J Comp Neurol, 411(1): 1-2

[114] Vex - A Visual Editor for XML [http://vex.sourceforge.net/]

[115] J.-D. Kim and J. Tsujii (2006), Corpora and their annotation, In, Text Mining for Biology and Biomedicine, Artech House: Boston. 179-212.

[116] L. W. Swanson and G. D. Petrovich (1998), Trends Neurosci, 21(8): 323-31

[117] L. W. Swanson (2000), Trends Neurosci, 23(11): 519-27

[118] S. Ananiadou, C. Friedman and J. Tsujii. (2004), Journal of Biomedical Informatics, 37: 393-395

[119] M. Krauthammer and G. Nedadic (2005), Journal of Biomedical Informatics, 37: 512-526

[120] L. W. Swanson (2004), Brain Maps: Structure of the Rat Brain. 3 ed. ed. Vol., San Diego: Elsevier Academic Press.

[121] O. Bodenreider (2006), Lexical, Terminological and Ontological Resources for Biological Text Mining, In, Text Mining for Biology and Biomedicine, Artech House: London.

[122] MiniPar home page [http://www.cs.ualberta.ca/ lindek/minipar.htm]

[123] S. Dingare, J. Finkel, M. Nissim, C. Manning and C. Grover (2004), A System For Identifying Named Entities in Biomedical Text: How Results From Two Evaluations Reflect on Both the System and the Evaluations. in 2004

BioLink meeting: Linking Literature, Information and Knowledge for Biology at ISMB 2004. http://nlp.stanford.edu/manning/papers/ismb2004.pdf.

[124] R. Rodriguez-Esteban, I. Iossifov and A. Rzhetsky (2006), PLoS Comput Biol, 2(9):

[125] J. Fleiss (1971), Psychological Bulletin, 76: 378-81

[126] C. Manning and S. H (1999), Foundations of Statistical Natural Language Processing. ed. ed. Vol.: MIT Press.

[127] D. Jurafsky and J. H. Martin (2000), Speech and Language Processing. An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. ed. ed. Vol., Upper Saddle River, NJ: Prentice-Hall, Inc.

[128] L. Hirschman, A. Yeh, C. Blaschke and A. Valencia (2005), BMC Bioinformatics, 6 Suppl 1: S1

[129] L. Adelman and S. L. Riedel (1997), Handbook For Evaluating Knowledge-Based Systems. ed. ed. Vol., Boston: Kluwer Academic Publishers.

[130] The NeuroScholar SourceForge Project Page [http://www.sourceforge.net/projects/neuroscholar]