Natural Language Processing and Systems Biology

K. Bretonnel Cohen and Lawrence Hunter

Center for Computational Pharmacology, University of Colorado School of Medicine, Denver, USA.
E-mail: {kevin.cohen, larry.hunter}@uchsc.edu

Summary. This chapter outlines the basic families of applications of natural language processing techniques to questions of interest to systems biologists and describes publicly available resources for such applications.

1 Introduction

Natural language processing (NLP) is the *processing*, or treatment by computer, of *natural language*, i.e., human languages, as opposed to programming languages. The two differ from each other in a very fundamental way: the interpretation of a programming language is designed not to be ambiguous, while the possible interpretations of natural language are potentially ambiguous at every level of analysis. The processing of computer languages is a subject for computer science and is generally treated in courses on compiler design. In contrast, the processing of natural language crosses a number of disciplines, including linguistics, computer science, and engineering.

One of the more surprising developments in bioinformatics and systems biology has been the attention that NLP has received at bioinformatics conferences in recent years. The Pacific Symposium on Biocomputing (PSB) and Intelligent Systems for Molecular Biology (ISMB) conferences began publishing papers on the topic in the early 1990s devoting entire sessions to the topic in the late 1990s. The natural language processing community has reciprocated, with the Association for Computational Linguistics offering workshops on NLP in the molecular and systems biology domain for the past three years ([56, 2, 48]). This is a welcome turn of events for the NLP community; although the medical domain has long been a target of interest for computational linguists, the medical community has yet to adopt natural language processing in a widespread way. In contrast, the current state of events, in which linguists find biologists coming to them, is a happy one. The results have been beneficial to both groups, with biologists gaining curation tools and linguists taking advantage of the large, well-curated resources that the biological community has made available in recent years. Biologists are increasingly faced with a body of literature that is too large and grows too rapidly to be reviewed by single researchers. At the same time, it becomes increasingly clear that relevant data is being published in communities outside of the traditional molecular biology subfields. Faced with the need to perform systematic surveys of all published information about multiple genes and proteins returned in large numbers by high-throughput assays, there is a growing awareness among molecular biologists that automated exploitation of the literature may be not just useful, but essential[1].

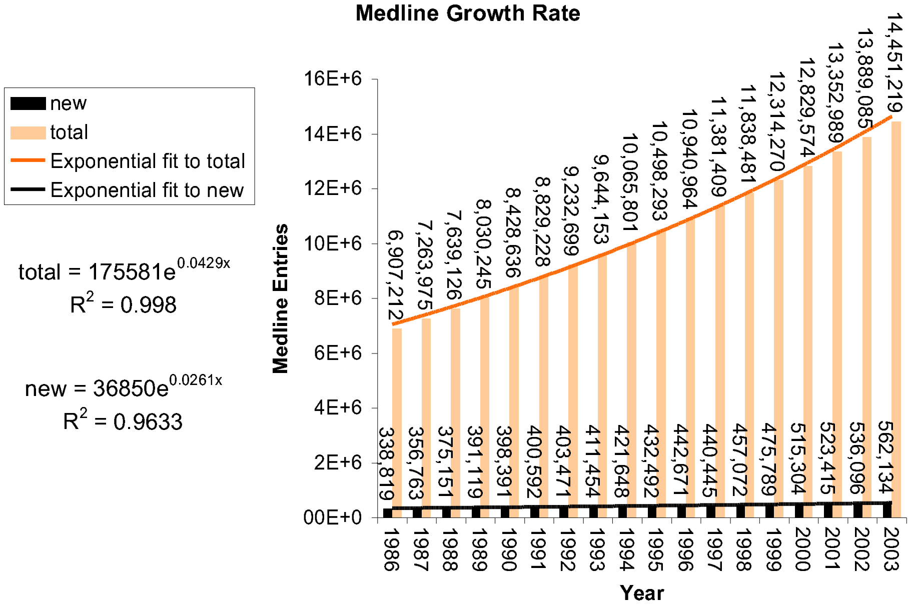

Fig. 1. Growth in Medline over the past 17 years. The hollow portion of the bar is cumulative size up to the preceding year; the solid portion is new additions in that year.

Unfortunately for impatient consumers—perhaps fortunately for curious scientists—NLP is approximately as difficult as it is important. It requires enormous amounts of knowledge, on a number of levels. For example, knowl-

[1] NLP techniques have also proven useful for macromolecular sequence analysis. This includes the use of hidden Markov models (see e.g., [57] for a general overview and [5] for biological applications), a technique from speech recognition; and the use of phrase structure grammars (see [97] for an excellent review and [96] for a more detailed exposition). In a related vein, techniques from computer science for efficient string searches and tree manipulations have been important as well; these are covered very well in [42]. These are beyond the scope of this chapter, and the reader is referred to the cited references for more information on them.

edge of how words are formed (*morphology*) is required to understand words like *deubiquitinization* that are complex and may not have been seen before. Knowledge of how phrases combine (*syntax*) is needed to understand why a sentence like *These findings suggest that FAK functions in the regulation of cell migration and cell proliferation* ([41]) is ambiguous (does FAK play a role in cell proliferation and in the regulation of cell migration, or does it play a role in the regulation of cell proliferation and in the regulation of cell migration?). These problems are difficult enough—despite the fact that since the 1960s most linguists have been working on the English language, there is still no comprehensive analysis of the syntax of English available. However, they pale next to the difficulty of representing the knowledge about the world that we make use of in understanding language. As human users of language, world knowledge is simultaneously so pervasive and so taken for granted in our understanding of language that we are generally unaware of it and may be difficult to convince that it plays a role at all. Consider the sentences *she boarded the plane with two suitcases* and *she boarded the plane with two engines* ([54]). Both sentences are equally syntactically ambiguous, with two possible phrasal structures for each sentence, depending on whether the plane or the woman has the suitcases or the engines. However, humans are unlikely to entertain two analyses of either sentence—rather, one analysis (and a different one in each case) seems obvious and exclusive for each. This phenomenon is based on knowledge that humans have about the kinds of things that people and airplanes are and are not likely to have. Representing this level of knowledge in the breadth and depth that would be required for understanding unrestricted text in general English or any other language has so far remained an elusive goal.

2 NLP and AI

NLP has a long history in artificial intelligence, and vice versa. There are two main lineages in NLP, one of which traces itself back to Roger Schank (see e.g., [93]). Historically, approaches to conceptual analysis work in AI have tended to be heavily based on semantic processing and knowledge-based techniques, with syntax often (although certainly not always) having a marginal position and sometimes eschewed completely, at least in principle. [103] provides a comprehensible overview of NLP research in the Schankian paradigm through the early 1980s. Since that time, two main trends of thought on NLP have emerged from the AI community: the *direct memory access* paradigm and implementations of conceptual dependency parsing. The *direct memory access* paradigm ([87, 69, 31]) has investigated the implications of thinking of language processing as change in mental representations. This offers some real advantages with respect to mapping the output of NLP to known concepts and entities in an ontology or knowledge base that is lacking in term-based approaches to NLP. (For example, for the input *...Hunk expression is restricted to subsets* *of cells...* ([38]), a good term-based system will output the fact that *Hunk* is expressed; a concept-based system might output the fact that LocusLink entry 26 559 is expressed.) It has had some commercial application in the area of robot control by NASA, and shows promise for language processing in the systems biology domain, including in the areas of word sense disambiguation and resolution of syntactic ambiguity ([53]). The conceptual dependency parser ([61, 88]) has had success in the kinds of information extraction tasks that are of much current interest in the bioinformatics community. (Other AI researchers have increasingly pursued statistical approaches to NLP, e.g., [19, 20]). Systems biology literature shows every indication of being the right target at the right time for AI-inspired approaches. Though the necessity of incorporating knowledge into language processing has long been acknowledged, in the past knowledge-based approaches have been thought to be impractical due to both the high cost of knowledge engineering and the breadth and depth of 'common-sense' knowledge required to parse general English. Within just the recent past, the cost argument has ceased to hold as much weight, as the molecular and systems biology community has released for public use large, carefully curated resources like LocusLink ([67] and the Gene Ontology ([24, 25]). With respect to the depth and breadth of knowledge required, we maintain that it is substantially less for molecular biology literature than for general English: nothing you need to know to understand molecular biology is everyday, common-sense knowledge- –everything that anyone knows about molecular biology came from a textbook, a journal article or an experiment. Thus, the time is ripe for applying AI to NLP in systems biology[2].

3 NLP and systems biology

The importance of NLP for systems biology comes from the high-throughput nature of modern molecular biology assays. The drinking-from-a-firehose nature of the business creates the opportunity for fruitful application of NLP techniques in two ways:

- It makes automated techniques for handling the literature attractive by fueling a rate of publication that is unequaled in the history of science, or indeed of the world.
- At the same time, it makes progress in the field of NLP possible by providing a huge body of data in a restricted domain for training and evaluation of NLP systems.

[2] It should be noted that the molecular biology domain has long been known to be a good target for natural language processing applications due to its good fit to a *sublanguage* model of language. A *sublanguage* is a genre of language use which deals with a semantically restricted domain and has certain qualities which make it more amenable to machine processing than is unrestriced language. A number of recent papers [33] have discussed the fit of systems biology texts to the sublanguage model.

Specific applications of NLP to biological data or assays include automated literature searches on sets of genes returned by an experiment; annotation of gene lists with Gene Ontology concepts; improvement of homology search; management of literature search results; aids to database curation; and database population. These biological tasks have been approached through a variety of NLP techniques, including information extraction, bibliometrics, and information retrieval. In addition, there are subtasks whose successful accomplishment is key to all of these. These include entity identification (see Section 4.2), tokenization (see page 14), relation extraction (see Section 4.3), indexing (see page 20), and categorization and clustering (see page 21). These subtasks are discussed in detail in the sections that follow.

3.1 Where NLP fits in the analysis pipeline

NLP fits into the bioinformatics data analysis pipeline in two ways, or at two points in the process: at the beginning, by aiding in the analysis of the output of high-throughput assays, thus helping the scientist bring a project from experiment to publication; and at the end, by helping the working researcher exploit the flood of publications that fills Medline at the rate of 1500 abstracts a day. (This split in times of application of the technology does not, however, correspond to any division of natural language processing techniques into different categories; as we will see, a given biological application can be implemented using a variety of different NLP technologies, and a single NLP technique may be utilized for a variety of types of biological applications.) We can also think of NLP techniques as helping the biologist approach two kinds of tasks: on the one hand, ad hoc location of data about single items of interest to them (where *single* might be a single gene or protein, or the output of an experiment, which might itself be a list, for example of differentially expressed genes). In this case, the strength of NLP is its potential for mining information from communities whose literature the researcher might not be aware of but whose work has relevance to her (consider for instance the newly discovered importance of a pregnancy-related protein in heart failure [29]). The other type of task can be thought of as systemic in nature, for example population of databases or aiding database curators; here we make more sweeping queries, of the nature of *tell me about every protein-protein interaction described in the literature.*

3.2 Database population and curation

Rapid population of databases of biologically interesting information was an early motivation for NLP in bioinformatics. The idea that if protein names could be located in text, then we could automatically populate databases of facts about proteins—for example, their interactions with other proteins, as in DIP and BIND—comes up in the first of the modern papers on molecular biology NLP, [35]. The problem to be solved is that enormous amounts of information on the topic are present in the systems biology literature, and we would like to convert that free-text information into a computable form, i.e., entries in structured databases. This is doable manually, but at great cost in terms of time and financial resources. Two basic approaches have been suggested—bibliometric techniques, and information extraction techniques.

Database population belongs to a class of problems in which the goal of NLP is to discover a very limited range of types of facts —perhaps only one. A typical example is protein-protein interactions. Bibliometric approaches are based on the assumption that if two proteins are mentioned in the same text (typically an abstract), then there might be a relationship between them. The PubGene system ([55]) is a good example of such a system. Sophisticated approaches like PubGene attempt to normalize for the fact that two proteins might be mentioned in the same text by chance. They typically find only pairwise interactions, an exception being the AlcoGene module of the INIA web site, which finds interactions of arbitrarily large arity. In general, bibliometric approaches suffer from problems related to entity identification. Either they are restricted with respect to the kinds of entity referents that they find—for example, PubGene utilizes only gene symbols and single-word names—or they are swamped by false positives due to synonymy issues, or both. [116] has a good discussion of sources of false positives in bibliometric approaches.

Information extraction offers a more constrained approach to database population. Examples of papers whose stated goal is database population using information extraction techniques include [8, 26]. Information extraction targets a very restricted set of types of assertions, and hence is less susceptible to the extreme low precision problems of bibliometric systems. In general, no technique has proven sufficiently accurate for completely automated population of databases. However, a number of techniques produce output that is of sufficient quality to aid human curators of such databases. Systems biology databases that store information that is more abstract than sequence data, such as BIND, Swiss-Prot, and OMIM, are typically hand-curated by experienced scientists, with new entries coming from findings reported in the scientific literature. A growing body of work addresses the needs of such curators for a fast and efficient way to navigate or filter the high volume of publications that characterizes the rapid rate of progress in systems biology today. The potential utility of NLP in curation efforts is so apparent that some recent competitions have been funded or materially aided by various databases.

3.3 Aids to analysis of high-throughput assays

Gene expression arrays

A number of studies have specifically addressed issues in the analysis of gene expression array data. An early such system was MedMiner ([107]), which was designed to perform automatic literature searches on large numbers of genes found to be of significance in an expression array study. Such studies often result in large lists of genes which may lead to thousands of articles being returned by a literature search; MedMiner helps the experimenters navigate these large bodies of literature by sorting them according to categories known to be of interest to molecular biologists. This work has been extended to other user communities, including researchers on the molecular biology of substance abuse and cancer researchers. Shatkay et al. [98] describes a method for detecting functional relationships in microarray data. Other approaches to the application of NLP to the interpretation of gene expression arrays have concentrated on using literature to augment the classifications of genes already present using the Gene Ontology ([86]).

3.4 Interaction and pathways

A significant body of work has concentrated on the discovery of networks of interactions and on pathway discovery. The interactions are generally between proteins, although other kinds of 'interactions' or associations have been investigated as well, including:

- Proteins and drugs ([90, 107])
- Proteins and diseases ([26, 100, 101])
- Proteins and subcellular locations ([26, 100, 101])

In general, the linguistic and computational problems are the same, regardless of the exact nature of the interaction.

4 Issues and resources in natural language processing

4.1 Evaluation

Metrics

Most evaluation in NLP is done by calculating values for precision, recall, and often F-measure on the output of a system, evaluated against a gold standard. Gold standard data is, in the best-case scenario, data that is hand-annotated by domain experts. It is often constructed by running some automated system against a set of inputs, and then having it hand-corrected by domain experts. It is preferable to have multiple human experts annotate the data. When this is done, inter-annotator agreement can be calculated, e.g., by calculating the κ statistic. This can be an indicator of the difficulty of the task, e.g., indicating the possible upper limit of system performance. Preparing such gold standard data sets is a pressing issue in the systems biology NLP domain. When gold standard data sets are available, they are listed in the relevant subsections.

Precision measures how often the system is correct when it outputs a particular value. It is similar to specificity, and is calculated by dividing the number of correct outputs (*true positives*, or *TP*) by the total number of outputs. The total number of outputs is the number of correct outputs plus the number of incorrect outputs (*false positives*, or *FP*), so the equation is often given as *P = TP/(TP + FP). Recall* measures how often the system correctly finds the right things to output. It is similar to sensitivity, and is calculated by taking the ratio of correct outputs by the total number of potential correct outputs. The total number of potential correct outputs is the number of correct outputs plus the count of things that should have been output but were not (*false negatives*, or *FN*, so the equation is often given as *R = TP/(TP + FN).* The *F-measure* or *harmonic mean* attempts to balance the contributions of precision and recall to system performance. It is calculated by *2PR/(P+R).*[3] [54] provides a cogent overview of these and other metrics for evaluating NLP systems.

Precision and recall are taken from the information retrieval (IR) community. The prototypical IR task is to retrieve from some set of documents all and only the documents that are relevant to some query. We assume that the set includes some documents that are relevant, and some that are not. Documents that are relevant and are successfully retrieved by the system are thus 'true positive' outputs. Documents that are retrieved by the system but that actually are *not* relevant to the query are 'false positive' outputs, and documents that truly are relevant to the query but that the system failed to retrieve are 'false negative' outputs.

Bake-offs

Most systems currently described in the literature were evaluated with locally prepared data, and sometimes with idiosyncratic scoring methods. However, in recent years the systems biology NLP community has experimented with its first 'bake-off'-style competitions, where each participating group is evaluated on the same data, with outputs being scored at a central location using consensus criteria. The recent competitions have been:

- the KDD Cup genomics challenge, described in [119]
- the TREC 2003 genomics track, described in [47]
- BioCreative, described at [6]

4.2 Entity identification

All applications of NLP to problems in computational and systems biology require the ability to recognize references to genes and gene products in text. For example, in the sentence fragment *association of ADHD with DRD4 and DRD5* ([60]), we want to know the DRD4 and DRD5 are genes, but ADHD is

[3] The F-measure can be calculated in other ways that allow for weighting precision more or less, relative to recall—see [57], and [68] pp. 268–270.

not, despite the fact that all three words look very much the same. The general problem of recognizing things of a particular class in free text is known as *entity identification* or *named entity recognition*. The problem was first defined in the general-language domain in the context of the Message Understanding Conferences ([78, 54]). Entity identification has been a topic of interest in the systems biology NLP domain for about as long as NLP has been of interest to systems biologists, and in fact the most heavily cited NLP paper from a computational bioscience conference, [35], was on this topic. In general-language domains, the set of entities has tended to be fairly heterogeneous, ranging from names of individuals to monetary amounts; in the systems biology domain, the set of entities is sometimes restricted to just genes and gene products, with individual authors tending to define the task on an ad hoc basis. (The BioCreative competition may bring about some standardization of the task definition.)

Approaches to entity identification in the systems biology domain fall into two general classes: rule-based approaches, and machine-learning-based approaches (see below). Rule-based approaches generally rely on some combination of regular expressions (see paragraph below) to define patterns that match gene names, and some logic for extending names to the right and/or left. For example, a rule-based approach might use a regular expression such as `/^[a-z]+[0-9]+$/` (any sequence of one or more lower-case letters followed immediately by any sequence of one or more digits) to recognize that p53 is a gene name. It might also include a rule (possibly also implemented as a regular expression) to include the word gene if it occurs immediately to the right of a string that is recognized by that pattern. In addition to Fukuda et al.'s work, examples of rule-based approaches in the literature include [79]. Fukuda's PROPER system is a rule-based system that is freely available for download at [58].

A variety of machine-learning-based approaches to entity identification have been tried. These are mostly the work of the Tsujii lab. Approaches have included decision trees, Bayesian classifiers, hidden Markov models, iterative error reduction, and support vector machines. Tanabe and Wilbur's ABGene system is a learning-based system that is freely available for download at [1].

Almost all work on entity identification can be described as entity 'location.' The task is generally defined as locating entities in text. There is generally no attempt to map the entities that have been located to a database of genes or gene products, despite the benefits to being able to do this. This more complex task may be referred to as concept identification. [21] addresses some of the problematic issues for this task from a structural linguistic perspective. BioCreative task 1B addressed the issue.

Entity identification systems that attempt to rely solely on 'look-up' in a dictionary or gazetteer of names typically perform quite poorly, with coverage generally only in the range of 10-30%, meaning that only 10-30% of the gene names in a corpus can typically be found this way, even allowing for some variability in the form of the names between the reference source and the corpus, such as letter case, hyphenation , etc. (see [21] for a discussion of such variability).

A *regular expression* is a mathematical formula for specifying the class of objects that belong to a particular set. When applied to natural language processing, the objects are textual strings. Regular expression engines typically allow for making reference to specific positions in a string, for allowing choices between a set of characters, for repetition, and for optionality. For example, in the regular expression `/∧[a-z]+[0-9]+$/` the carat species the beginning of the string, `[a-z]` represents a choice between any of the lower-case letters, `[0-9]` represents a choice between any of the digits, the plus-signs indicate that the ‘choice’ that precedes it can be repeated any number of times, and the dollar-sign specifies the end of the string. Taken together, the elements of the regular expression specify strings that begin with one or more letters and end with one or more digits. Thus, the set of strings that is specified by the regular expression includes *p53, pax9,* and *hsp60.* Chapter 2 of [57] gives an excellent introduction, both theoretical and applied, to regular expressions. [50] provides an excellent introduction to the use of regular expressions in the Perl programming language; because most modern regular expression engines mimic Perl’s syntax, much of its material is applicable to other languages as well.

Resources for entity identification

Resources for entity identification fall into two classes:

- Lists of names and symbols for ‘dictionary’ construction
- Software for performing entity identification

At this writing, a variety of publicly available sources for dictionary construction exist. Most provide both names and symbols, and some also provide synonym lists. These include the following:

- LocusLink’s LL_tmpl file. LocusLink ([67]) supplies an extremely large number of names and symbols, including synonyms for each, often multiple ones. These are available for a wide variety of species (thirteen at time of writing). There is no attempt at standardization. From the point of increasing recall, this is a benefit. Names and symbols can be extracted from a number of fields, including
  - OFFICIAL_GENE_NAME
  - PREFERRED_GENE_NAME
  - OFFICIAL_SYMBOL
  - PREFERRED_SYMBOL
  - PRODUCT
  - PREFERRED_PRODUCT
  - ALIAS_SYMBOL
  - ALIAS_PROT

It is available for downloading at [65]. Java classes for parsing the data file and representing LocusLink entries are available from the authors.

- HUGO: The Human Gene Nomenclature Database supplies a much smaller number of names and symbols for human genes. Some symbols are provided. The symbols are standardized. It is described in [114] and is available for downloading in a variety of formats at [51].
- FlyBase provides names, symbols, and synonyms for D. melanogaster genes. [49] and [77] discuss its use in NLP.

Finally, the reader should consult the ‘interesting gene name’ site at [39]; for comic relief, be sure to note the ‘worst gene names’ page. See also the FlyNome site (`http://www.flynome.org`) for explanations of some of the more interesting Drosophila names.

Software for performing entity identification falls into two classes—systems that are available over the Internet for remote usage, and systems that the user installs locally. Availability of the former type of system of course varies. At the time of writing, the following over-the-Internet systems are available:

- The GAPSCORE system is available via a Web-based interface at [37]. It returns a list of potential gene names in the input text with a score that rates the probability that each name is a gene. (It is also available through an XML-RPC interface from a variety of languages—see below.) It is described in [18].
- Yapex is available via a Web-based interface at [118]. Yapex has the unusual feature of being able to use information about names mentioned more than once in an input to improve recognition of those names on subsequent mentions. It is described in [32].
- The Descriptron system, under development at the Center for Computational Pharmacology and described in [74], provides (among other services) look-up for gene symbols, names, and other identifiers, allowing rapid determination of whether or not an identifier is ambiguous as to species or as to the underlying sequence data.

The following systems for local installation are available:

- Chang et al.’s GAPSCORE system is available at [36].
- The ABGene system is available for download at [1]. It performs two functions at once: it simultaneously locates named entities, and performs part-of-speech tagging, such that all non-entities in the output have POS tags in place. It is available on Solaris and Linux; installation on Linux requires Slackware (a specific Linux distribution, available at [102]). It is described in [108] and [109].
- The KeX/PROPER system is available for download at [58]. It produces SGML-style output. It is optimized for yeast. It is described in [35].

Evaluation of entity identification

Two kinds of data sets for evaluation of entity identification systems exist. One kind is data sets assembled by individual authors and made available in conjunction with their publications. Recently, another kind of data set has become available, as well—publicly available, carefully-curated large data sets intended for use in challenge tasks. These latter may become standard data sets for publication purposes.

- The GENIA corpus is an extensively hand-annotated corpus of abstracts on human blood cell transcription factors. It is split into sentences and the content is fully tokenized[4]. It is part-of-speech tagged, and is also annotated with respect to a sophisticated ontology of the molecular domain. This ontology includes a number of concepts that correspond to named entities as that term is used in this chapter, i.e., genes and gene products. It is the largest corpus of its type currently available, comprising 2 000 abstracts with 18 545 sentences containing 39 373 named entities. It is available at [40] and is fully described in [81, 59].
- The BioCreative corpus comprises 10 000 sentences and titles with 11 851 named entities. Unlike the GENIA corpus, it was deliberately constructed to be heterogeneous (within the constraints of the molecular biology domain). It includes sentences that contain deliberately challenging false positives. It is downsampled from abstracts, which removes some classes of contextual cues. The corpus was originally constructed by the National Library of Medicine. It was made publicly available in conjunction with the BioCreative comptetition on entity identification. It is available for download at [6].
- the Yapex data set—about 200 Medline abstracts, some of which are a re-annotated subset of the GENIA corpus. It is available for download at [117].
- The authors make available a system for generating test data for entity identification systems at [52]. The system allows the user to generate customized test suites to evaluate performance on different types of names and symbols in a variety of sentential contexts. It is described in [23].

4.3 Information extraction

Information extraction (IE) is the location of assertions about restricted classes of facts in free text. It is also sometimes referred to as relation extraction. IE can be thought of as a ‘robust’ approach to natural language

[4] *Tokenization* is the separation of input into appropriately sized chunks for analysis. The term often refers to separating words and punctuation into individual *tokens* (see the example on page 15). *Sentence tokenization* is the separation of input into sentences.

understanding ([57]) in that rather than trying to build a system that 'understands' all aspects of an input text, workers in information extraction try to 'understand' only assertions of a very restricted sort. For example, an early system in the molecular biology domain extracted assertions about subcellular localization of proteins ([26]). Information extraction technologies have a wide range of applications. The most basic of these uses the results of information extraction directly to populate a knowledge base. Extracted assertions can also be used as input data for other NLP-based applications, such as ontology construction, network discovery, and information retrieval. (So far the immaturity of the technology has stood in the way of success in such efforts.)

Approaches to information extraction can be classified into two broad categories—rule-based, and machine-learning-based. In general, rule-based systems tend to apply some linguistic analysis; in contrast, learning-based systems tend to apply less linguistic analysis and to use simpler representations[5]. The first application of information extraction to the molecular biology domain was a rule-based system for finding protein-protein interactions, described in [8]. A representative example of a rule-based system is described in [83]. These authors developed a set of regular expressions defined over part-of-speech (POS) tags and entities that perform some analysis of sentence structure, such as recognizing complex coordinated sentences, and then recognize simple assertions about protein-protein interactions involving a limited number of verbs and deverbal nouns. Commonly used 'keywords' in these systems (see e.g., [8, 10]) include:

- *interact*
- *associate*
- *bind*
- *complex*
- *inhibit*
- *activate*
- *regulate*
- *encode*
- *function*
- *phosphorylate*

The first machine-learning-based information extraction system in the molecular biology domain is described in [26]. They developed a Bayesian classifier which, given a sentence containing mentions of two items of interest,

[5] A very common model for representing a text in a machine learning framework is the *bag of words* (BOW). In a BOW model, the text is represented as a vector in which each element represents a single word. (The value for the element may be binary, i.e., indicating presence or absence of the word, or it may be weighted in some way.) The BOW metaphor takes its name from the fact that the features reflect nothing but the words, crucially excluding order—thus, the BOW representation for the sentences *A upregulates B* and *B upregulates A* would be identical—something like *A:1 B:1 upregulates:1.*

returns a probability that the sentence asserts some specific relation between them. For example, given a sentence containing the name of a protein and the name of a cellular compartment, it returns the probability that the sentence asserts that that protein is localized to that cellular compartment. Later systems have applied other technologies, including hidden Markov models and support vector machines.

Things that make information extraction difficult

A variety of factors conspire to make information extraction difficult. These factors fall into two general groups: issues that must be dealt with in most information tasks, and issues that may be specific to the systems biology domain. Entity identification is frequently cited in error analyses as a source of low recall: inability to solve the entity identification problem leads to missed assertions. *Coordination,* the linking of structures by words like *and* and *or,* is another problematic phenomenon. Negation is often simply ignored, a notable exception to this being the work reported in [62] and [63]. *Anaphora,* or references to entities that have been named earlier, often by words like *it,* are another source of low recall.

Low-level linguistic analysis and preprocessing

Many issues of low-level linguistic analysis arise in information extraction. These include:

- sentence tokenization
- word-level tokenization
- entity identification
- part-of-speech tagging
- stemming
- abbreviation expansion

The National Library of Medicine makes available a variety of tools that might be of use. These include:

- *lvg* (lexical variant generation), a set of Java API's for normalizing and generating variant forms of biomedical terms. lvg is described in [28] and is available for download at [66].
- *MetaMap*, a system for finding biomedical concepts in free text. MetaMap is described in [3] and is available for download at [76].

Sentence tokenization

Sentence tokenization is the process of separating a chunk of text into individual sentences. A problem with tokenization of sentences in molecular biology text is that case is not always a reliable indicator of sentence boundaries. Consider for example the following text, which should be split into four sentences, indicated here by line breaks:

> *Misshapen (Msn) has been proposed to shut down Drosophila photoreceptor (R cell) growth cone motility in response to targeting signals linked by the SH2/SH3 adaptor protein Dock.*
> *Here, we show that Bifocal (Bif), a putative cytoskeletal regulator, is a component of the Msn pathway for regulating R cell growth cone targeting.*
> *bif displays strong genetic interaction with msn.*
> *Misshapen (Msn) has been proposed to shut down Drosophila photoreceptor (R cell) growth cone motility in response to targeting signals linked by the SH2/SH3 adaptor protein Dock.*

The final sentence of the selection (from the abstract of [91]) begins with a mention of the recessive form of the Bifocal gene. The authors have followed the Drosophila community's convention of indicating dominance/recessiveness of an allele by using upper case for the initial letter of the name/symbol when discussing the dominant form, and lower case for the recessive allele. Other difficulties come from domain-specific entities that can contain internal punctuation that would normally be sentence-final, such as chromosomal locations (*p24.2*), species names (*S. cerevisiae*), etc.

Approaches to sentence tokenization can be divided into two categories: rule-based, and learning-based. Appendix B of [16] gives a set of heuristics for rule-based sentence tokenization of Medline abstracts. No publicly distributed tools that are customized for the molecular biology domain are currently available.

Word-level tokenization

Most NLP projects require breaking the input into word-sized chunks. The definition of what counts as a *word* is often frustratingly domain-specific. Molecular biology text provides its own challenges in this regard. For example, tokenization routines generally split punctuation from the words to which it is attached. They generally count hyphens as separable punctuation. However, this often yields undesirable results on molecular biology text. Consider, for example, the sentence *Relaxin, a pregnancy hormone, is a functional endothelin-1 antagonist: attenuation of endothelin-1-mediated vasoconstriction by stimulation of endothelin thp-B receptor expression via ERK-1/2 and nuclear factor-kappaB* [29]. The desired output of tokenization is shown in Table 1, where it is contrasted with the output of a typical tokenizer. A number of problems with the typical output are apparent. *Endothelin-1-mediated* should be separated into *endothelin-1* and *mediated*, but *endothelin-1* should be kept as a single 'token.' Similarly, *thp-B* is split into three tokens, when it should be maintained as a single unit, and *ERK-1/2* is split into five units. Some tokenization routines actually discard punctuation, including hyphens; this is problematic in the biomedical domain, where e.g., hyphens can be used to indicate negation ([105]) and electrical charge.

Part-of-speech tagging

Part-of-speech tagging is the assignment of part-of-speech labels to individual tokens in a text. The set of labels is typically much larger than the eight categories (noun, verb, preposition, etc.) typically taught in traditional grammar. A common *tagset* (set of tags) includes around forty categories, and much larger sets are known, as well. ([57]:Appendix C gives several pages of tags.) The increased size of NLP tagsets as compared to the eight traditional parts of speech comes in part from finer granularity—for example, where traditional grammar has the category *noun*, a commonly used NLP tagset has the categories *NN (singular or mass noun), NNS (plural noun), NNP (singular proper noun),* and *NNPS (plural proper noun).* It is a challenging task because even within a homogeneous domain, a word can have multiple parts of speech. For instance, in the molecular biology domain, *white* can be an adjective, as in *...Morgan's awarenesss that white eye-color was not the only genetically determined alternative to red eye-color...* ([30]); a mass noun, as in *...the appearance of a traite, such as color, was due to the presence of a gene, and white, i.e., no color, to its absence* (op cit); and of course a proper noun. Information extraction systems generally apply a POS tagger and entity identification system as their first steps, in one order or the other. Publicly available entity identification systems are discussed above in 4.2. Publicly available POS taggers include the following:

- Brill: the Brill part-of-speech tagger ([14]) is possibly the most widely used piece of NLP software in the world. It is shipped with data for tagging general English, but can be trained on molecular biology data and has been widely applied to such.. It is available at [13].
- TnT: The *Trigrams'n'Tags* part-of-speech tagger ([12]), also known as *TnT*, is a very fast and stable part-of-speech tagger that is available on a variety of platforms. It has been tested on multiple languages, and has an intuitive interface. It is available at [111].

Stemming

It is often useful to be able to determine the stem of words in the input text. A word's *stem* is the main part of the word, exclusive of parts that are added to it to mark plurality, tense, etc. For example, *interact* is the stem of the words *interacts, interacted, interacting,* and *interaction.* Publicly available software for this includes many implementations in a wide variety of languages of the Porter stemmer ([85]), available at [84]. No stemmer has been optimized for NLP in the systems biology domain.

Table 1. Desired and typical outputs of tokenization. The table shows one token per line. Note that the typical tokenization routine tends to break apart things that should remain single units.

Desired output of tokenization	Output of a typical tokenization routine
Relaxin	Relaxin
,	,
a	a
pregnancy	pregnancy
hormone	hormone
,	,
is	is
a	a
functional	functional
endothelin-1	endothelin
	-
	1
antagonist	antagonist
:	:
attenuation	attenuation
of	of
endothelin-1	endothelin
	-
	1
-	-
mediated	mediated
vasoconstriction	vasoconstriction
by	by
stimulation	stimulation
of	of
thp-B	thp
	-
	-
	B
receptor	receptor
expression	expression
via	via
ERK-1/2	ERK
	-
	1
	/
	2
and	and
nuclear	nuclear
factor-kappaB	factor
	-
	kappaB
.	.

Lexical resources

Lexical resources are often useful in information extraction, and happily, a number of them are publicly available. (One could argue that the number and high quality of lexical resources that has become available in the recent past make molecular biology the first domain in which knowledge-based NLP has ever been practical.) The advantages of these resources include the ability to recognize multi-word terms, which reduces the amount of low-level parsing necessary ([115]).

- Gene Ontology: the Gene Ontology ([24, 25]) is an ontology of concepts relevant to the systems biology domain. [73, 113, 80] discuss various linguistic aspects of the ontology and its applicability to NLP tasks.
- UMLS: the Unified Medical Language System is a large metathesaurus of biomedical vocabularies. It is documented in [64] and Bodenreider (2004). and [11]. It is available through the National Library of Medicine at [112]. Numerous researchers have investigated its use in natural language processing, including [70, 71, 89, 4, 120, 15, 46, 3, 72], to name just a few.

Abbreviation expansion

The ability to deal with abbreviations is often important in systems biology text. Abbreviations are often defined ad hoc, limiting the usefulness of dictionary-based systems. Additionally, systems biology text also often contains gene symbols. These symbols are often defined in the text in a structure similar to that of an abbreviation definition.

- The BioText project makes Java code for a rule-based system available at [94]. It is straightforward to implement and use, and the authors and others have applied it to the BioCreative Task 1A challenge task. The algorithm is described in [95].
- Chang et al. make a statistically-based system available through a web site and via an XML/RPC server. It can be found at [104]. This system returns a list of potential abbreviation/definition pairs, each with both a categorical and a probabilistic assessment of the likelihood that it is a valid pair. The system is described in [17].

Evaluation of information extraction

There has not yet been a MUC-like competition for information extraction in the molecular biology domain, and so no data set like BioCreative exists yet. Small, generally ‘lightly annotated’ data sets have been made available by individual researchers. These include:

- Contact Mark Craven at `craven@biostat.wisc.edu` for access to a large dataset of assertions about protein-protein interactions, protein-disease associations, and subcellular localization of proteins.

- Contact Christian Blaschke at `blaschke@cnb.uam.es` for access to a dataset on protein-protein interactions.

Evaluations of information extraction in this domain typically involve precision, recall, and F-measure, but may differ with respect to the domain over which they are measured. Some authors calculate them on the basis of mentions in text, while other authors calculate them on the basis of the underlying concepts. For example, if a test set contains three assertions to the effect that p27 interacts with CK2, then if we are calculating recall on the basis of mentions, then there are three potential true positives, and any that we miss will count as false negatives. On the other hand, if we are calculating recall on the basis of the underlying concepts, then as long as we find at least one assertion that p27 interacts with CK2, we have no false negatives.

The issue of input size

Most NLP work in systems biology takes abstracts (with their titles) as the basic unit of input, rather than full-text articles. One reason for this is purely practical—until recently, access to full-text articles in easily processable formats has been quite limited. (The PubMed Central collection is one current attempt to make full-text articles available.) A small number of researchers has in fact reported success in working with full-text articles. The GENIES system ([34]) was evaluated on a full-length article, and the winning team in the 2002 Genomics KDD cup [119] used full-text articles to great advantage. [109] discusses some of the difficulties that arise when working with full-length articles rather than abstracts, and [27] evaluates the use of inputs of various sizes in an information extraction task.

Resources: raw data for information extraction

Resources: almost all research in systems biology NLP begins with a query to the Entrez interface to the Pubmed document collection, typically through the well-known Entrez interface. Kevin Rosenberg makes LISP code for accessing PubMed available through the BioLisp organization (`http://www.biolisp.org`), and the National Library of Medicine makes an API available. These queries themselves fall into the category of information retrieval, the subject of the next section.

A local copy of medline allows for heavier usage and faster access than does the National Library of Medicine interface or API's. The BioText project at the University of California at Berkeley and Stanford makes available Java and Perl code for parsing the data files provided by the National Library of Medicine into a relational database, as well as the associated schemas. The code and schemas are available at [7] and are described in [82].

4.4 Information retrieval

Information retrieval consists of finding subsets of documents in a larger set that are relevant to some query. Originally a problem in library science, it has largely been reconceived as a WWW query task. In the systems biology domain, all of the standard IR problems present themselves. In addition, there is a twist that is peculiar to this domain. A typical Web query may return thousands of documents, of which only a small number are actually relevant to the user—the 'needle in a haystack' problem. In contrast, a typical query about a gene to the Pubmed search engine may return thousands of documents, of which most are relevant to the user. So, the problem in IR for gene expression array studies is not the Google-task of finding a needle in a haystack—the problem is that the whole haystack is made of needles. The issue then becomes: how to organize this mass of documents in such a way as to make it navigable by the user? Relevant research issues include indexing, query construction, clustering/categorization, and visualization, which are discussed in the following sections.

Indexing

Indexing a document collection is the process of determining the set of terms or words within each individual document that should be used when matching that document to a query. Not all words are equally useful for purposes of indexing. For example, *function words* (words that indicate grammatical information only) such as *a, the,* and textitall are usually considered not to be useful for indexing. Since all documents contain them, they are not useful for determining whether or not a particular document should be returned in response to a particular query. In contrast, *content words* (words that express specific semantic concepts), such as *phosphorylate, protein,* and *BMP-4*, are generally good candidates for indexing. Standard mathematical procedures for determing the usefulness of particular words for indexing particular document collections exist—see Salton 1989 and Jackson and Moulinier 2002). To understand why a particular word might be useful for indexing one document collection but not another, consider the word *protein* and two separate document collections: a set of documents about nutrition, and a set of documents about Bone Morphogenetic Protein 4. For the set of documents about nutrition, it is easy to imagine realistic queries for which the presence or absence of the word *protein* in a document will be very useful for deciding whether or not to include that document in the set of documents that are returned. In contrast, for the set of documents that are all about Bone Morphogenetic Protein 4, the presence or absence of the word *protein* is not likely to ever help us decide whether or not to return a particular document.

A number of factors conspire to make indexing for systems biology difficult. These include:

- Massive synonymy of the items of interest. Many of the concepts of interest in systems biology are genes and proteins, which have on average about five synonyms each.
- Multi-word units: Traditional indexing assumes that the unit of interest is a single word. However, the concepts of interest to systems biologists are frequently referenced by multi-word units—review of two corpora of molecular biology texts revealed that about 50% of the mentions of genes and proteins in each corpus were two or more words in length ([22]).
- Difficulty of mapping to a standard ontology: Concepts of interest in systems biology are not limited to genes and proteins, but rather include also concepts such as those described in the Medical Subject Headings (MeSH) and the Gene Ontology. Such ontologies have proven to be useful indexing schemes, but assigning the correct MeSH headings or GO codes is difficult to do automatically, due to synonymy and to variability in the possible forms of multi-word ontology elements. (For example, *regulation of cellular proliferation* can also appear in text as *cell proliferation regulation.*)

  All of these are open research issues.

Clustering and categorization

Clustering and categorization address the issue of taking a set of documents that have been returned in response to a query, and organizing them in a way that helps the user navigate and make sense of them. There are two approaches to clustering and categorization: top-down, and bottom-up.

- *Top-down* clustering organizes a document set according to a pre-existing model of how a user models the conceptual domain. For example, the AlcoGene system is a literature retrieval application for researchers in the molecular biology of alcoholism. It organizes the set of documents returned in response to a query according to which of the following topics they address:
  - nervous system structures
  - behaviors
  - ion channels
  - protein kinases
  - quantitative trait loci

  These categories were arrived at by interviewing domain experts and monitoring their interactions with the literature retrieval system. To see a top-down categorization system in action, try the MedMiner web site, described in [107] and available at [75].
- *Bottom-up* clustering of documents is based on similarities between documents in a set as determined by some metric. Where top-down clustering is based on a priori assumptions about the world to which we map the members of a document set, bottom-up clustering is based entirely on the contents of the documents and requires no model of the world beyond a

theory of how to represent document contents and a similarity metric by which to assess them. To see a bottom-up categorization system in action, try Vivisimo's system at `http://www.vivisimo.com`. Limiting the search domain to PubMed, enter the query *p53*. The search returns 200 documents, which Vivisimo separates into a number of categories, including the following:

- breast (13 documents)
- activation of caspase (12 documents)
- hepatocellular carcinoma (10 documents)

Querying with the gene symbol *bmp4* returns a very different set of categories, including:

- neural crest (32 documents)
- tooth (17 documents)
- Tgfbeta (16 documents)
- receptors, fibroblast growth factor (12 documents)

The clusters for the oncogene p53 and the developmental gene bmp4 are quite different. This flexibility is a strength of the bottom-up approach. On the other hand, the clusters are not necessarily relevant to the researcher's interests; the guarantee of relevance is a strength of the top-down approach. [98] presents another perspective on clustering for information retrieval, assuming a usage scenario involving large lists of genes as for example the output of an expression array experiment. Clusters based on papers that are prototypical for particular genes are used to discover functional relationships between genes in a large dataset.

Visualization

Visualization: A very open area of research is visualization of the contents of large document collections. The hypothesis is that users might better be able to navigate large document sets if they have some visual metaphor for the organization of the set, rather than just the flat (or at best hierarchical) lists returned by most literature search interfaces. A good starting point for research in this area is [44]. For a demonstration of an interesting visualization system, see the Pacific Northwest National Lab's ThemeRiver, documented in [43] and viewable at [110].

5 Further reading

In this section I differentiate between general natural language processing, i.e., coverage of the topic that is not specific to a particular genre or domain, and NLP for systems biology. Any investigation of general NLP should start with [57]. For the specific topics of general information extraction, information retrieval, text categorization, entity identification, and summarization, the reader should begin with [54]. [106] describes molecular-biology-specific information retrieval, entity identification, and information extraction systems. For a comprehensive treatment of statistical approaches to general NLP, see [68]. For information retrieval, [92] is a good general text, making up for the fact that it is somewhat dated by the fact that it is incredibly readable; for information retrieval in biomedical domains, [45] is recommended. For NLP in the systems biology domain, some excellent review papers are available, including [9, 116, 99].

Acknowledgments

The authors gratefully acknowledge editing by Todd Kester and help with LaTeX hacking from Sonia Leach.

References

1. ABGene. `ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe`.
2. S. Ananiadou and J. Tsujii, editors. *Proceedings of the ACL 2003 workshop on natural language processing in biomedicine.* Association for Computational Linguistics, Association for Computational Linguistics, 2003.
3. A.R. Aronson. Effective mapping of biomedical text to the umls metathesaurus the metamap program. In *Proceedings of the AMIA Symposium 2001*, pages 17–21, 2001.
4. A.R. Aronson and T.C. Rindflesch. Query expansion using the umls metathesaurus. In *Proc. AMIA Annu Fall Symp 1997*, pages 485–489, 1997.
5. P. Baldi and B. Søren. *Bioinformatics: the machine learning approach.* MIT Press, 2nd ed edition, 2001.
6. BioCreative. `http://www.mitre.org/public/biocreative/2`.
7. BioText. `http://biotext.berkeley.edu2`.
8. C. Blaschke, M.A. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Intelligent Systems for Molecular Biology 1999*, pages 60–67, 1999.
9. C. Blaschke, L. Hirschman, and A. Valencia. Information extraction in molecular biology. *Briefings in Bioinformatics*, 3(2):154–165, 2002.
10. C. Blaschke and A. Valencia. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, 17(2):14–20, 2002.
11. O. Bodenreider. Unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(D):D267–D270, 2004.
12. T. Brants. Tnt—a statistical part-of-speech tagger. In *Proc. of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, 2000.
13. Brill. Pos tagger site. `http://www.cs.jhu.edu/~brill`.
14. E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, 1995.
15. D.A. Campbell and S.B. Johnson. A technique for semantic classification of unknown words using umls resources. In *Proc AMIA Symp 1999*, pages 716–720, 1999.

16. J.T. Chang. *Using machine learning to extract drug and gene relationships from text.* PhD thesis, Stanford University doctoral dissertation., 2003.
17. J.T. Chang, H. Schütze, and R.B. Altman. Creating an online dictionary of abbreviations from medline. *J Am Med Inform Assoc*, 9(6):612–620, 2002.
18. J.T. Chang, H. Schütze, and R.B. Altman. Gapscore: finding gene and protein names one word at a time. *Bioinformatics*, 20(2):216–225, 2004.
19. E. Charniak. *Statistical language learning.* MIT Press., 1996.
20. E. Charniak. A maximum-entropy-inspired parser. In *Proc. of NAACL-2000*, pages 132–139, 2000.
21. K.B. Cohen, A.E. Dolbey, G.K. Acquaah-Mensah, and L. Hunter. Contrast and variability in gene names. In *of the workshop on biomedical natural language processing.* Association for Computational Linguistics., 2002.
22. K.B. Cohen, P.V. Ogren, S. Kinoshita, and L. Hunter. Entity identification in the molecular biology domain with a stochastic pos tagger. in preparation.
23. K.B. Cohen, L. Tanabe, S.Kinoshita, and L. Hunter. A resource for constructing customized test suites for molecular biology entity identification systems. In *Linking biological literature, ontologies and databases: tools for users*, pages 1–8. Association for Computational Linguistics, 2004.
24. Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature*, 25:25–29, 2000.
25. Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research*, 11:1425–1433, 2001.
26. M. Craven and J. Kumlein. Constructing biological knowledge bases by extracting information from text sources. In *Proc. of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, pages 77–86. AAAI Press, 1999.
27. J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining medline: abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing 7*, pages 326–337, 2002.
28. G. Divita, A.C. Browne, and T.C. Rindflesch. Evaluating lexical variant generation to improve information retrieval. In *Proc AMIA Symp. 1998*, pages 775–779, 1998.
29. T. Dschietzig, C. Bartsch, C. Richter, M. Laule, G. Baumann, and K. Stangl. Relaxin, a pregnancy hormone, is a functional endothelin-1 antagonist: attenuation of endothelin-1-mediated vasoconstriction by stimulation of endothelin thp-b receptor expression via erk-1/2 and nuclear factor-kappab. *Circ Res*, 92(1):32–40, 2003.
30. R. Falk and S. Schwartz. Morgan's hypothesis of the genetic control of development. *Genetics*, 134:671–674, 1993.
31. W. Fitzgerald. *Building embedded conceptual parsers.* PhD thesis, Northwestern University doctoral dissertation., 1995.
32. K. Franzén, G. Eriksson, F. Olsson, L. Asker, P. Lidén, and J. Cöster. Protein names and how to find them. *International Journal of Medical Informatics*, 67(1-3):49–61, 2002.
33. C. Friedman. Sublanguage—zellig harris memorial. *Journal of Biomedical Informatics*, 35(4):213–277, 2002.
34. C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky.
35. K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: identifying protein names from biological papers. In *Pacific Symposium on Biocomputing 1998*, pages 705–716, 1998.

36. GAPSCORE. Code for gapscore site. http://bionlp.stanford.edu/webservices.html.
37. GAPSCORE. Site to identify the names of genes and proteins. http://bionlp.stanford.edu/gapscore/.
38. H.P Gardner, G.B. Wertheim, S.I. Ha, N.G. Copeland, D.J. Gilbert, and N.A. Jenkins et al. Cloning and characterization of hunk, a novel mammalian snf1-related protein kinase. *Genomics*, 63(1):46–59, 2000.
39. Genenames. Clever gene names website. http://tinman.vetmed.helsinki.fi/eng/intro.html.
40. GENIA. Automatic information extraction from molecular biology texts. http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.
41. A.P. Gilmore and L.H. Romer.
42. D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, 1997.
43. S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: visualizing thematic changes in large document collections. *IEEE transactions on visualization and computer graphics*, 8(1):9–20, 2002.
44. M.A. Hearst. User interfaces and visualization. In In Baeza-Yates and Ribeiro-Neto, editors, *Modern Information Retrieval*, pages 257–324. ACM Press, 1999.
45. W. Hersh. *Information retrieval: a health and biomedical perspective.* Springer Verlag, 2nd ed edition, 2002.
46. W. Hersh, S. Price, and L. Donohoe. Assessing thesaurus-based query expansion using the umls metathesaurus. In *Proc AMIA Symp 2000*, pages 344–348, 2000.
47. W.R. Hersh and R.T. Bhupatiraju. Trec genomics track overview. In *The Twelfth Text Retrieval Conference—TREC 2003*, 2003.
48. L. Hirschman, editor. *Linking biological literature, ontologies and databases: tools for users.* Association for Computational Linguistics, 2004.
49. L. Hirschman, A.A. Morgan, and A. Yeh. Rutabaga by any other name: extracting biological names. *Journal of Biomedical Informatics*, 35:247–259, 2002.
50. P. Hoffman. *Perl for Dummies.* For Dummies, 4th ed edition, 2003.
51. HUGO. The download site of the human genome organisation. http://www.gene.ucl.ac.uk/nomenclature/code/ftpaccess.html.
52. Hunter. Entity identification test suite generation site. http://compbio.uchsc.edu/Hunter_lab/testing_ei/.
53. L. Hunter and K.B. Cohen. Using ontologies for text analysis. In *Sixth annual bio-ontologies meeting Intelligent Systems for Molecular Biology*, 2003.
54. P. Jackson and I. Moulinier.
55. T.K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28. PMID: 11326270.
56. S. Johnson. Proc. of the workshop on natural language processing in the biomedical domain. Association for Computational Linguistics, 2002.
57. D. Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition.* Prentice Hall, 2000.
58. KeX. Kex download site. http://www.hgc.ims.u-tokyo.ac.jp/service/tooldoc/KeX/intro.html.

59. J.D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus—semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl. 1):180–182, 2003.
60. V. Kustanovich, J. Ishii, L. Crawford, M. Yang, J.J. McGough, and J.T. McCracken et al. Transmission disequilibrium testing of dopamine-related candidate gene polymorphisms in adhd: confirmation of association of adhd with drd4 and drd5. *Molecular Psychiatry*, Molecular Psychiatry 2003. PMID 14699430.
61. W. Lehnert. Subsymbolic sentence analysis: exploiting the best of two worlds. In Barnden and Pollack, editors, *High-level connectionist models (Advances in neural and connectionist computational theory Volume 1*, 1991.
62. G. Leroy and H. Chen. Filling preposition-based templates to capture information from medical abstracts. In *Pacific Symposium on Biocomputing 2002*, pages 350–361, 2002.
63. G. Leroy, H. Chen, and J.D. Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of Biomedical Informatics*, 36:145–158, 2003.
64. D.A. Lindberg, B.L. Humphreys, and A.T. McCray. The unified medical language system. *Methods Inf Med*, 32(4):281–291, 1993.
65. LocusLink. Locuslink download site. `ftp://ftp.ncbi.nih.gov/refseq/LocusLink/`.
66. lvg. lvg (lexical variant generation) site. `http://umlslex.nlm.nih.gov/lvg/2003/index.html`.
67. D. Maglott. Locuslink: a directory of genes. In *The NCBI Handbook*, pages 19–1 to 19–16, 2002.
68. C.D. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, 1999.
69. C. Martin. *Direct Memory Access Parsing.* PhD thesis, Yale University doctoral dissertation, 1991.
70. A.T. McCray. Extending a natural language parser with umls knowledge. In *Proc Annu Symp Comput Appl Med Care 1991*, pages 194–198, 1991.
71. A.T. McCray, A.R. Aronson, A.C. Browne, T.C. Rindflesch, A. Razi, and S. Srinivasan. Umls knowledge for biomedical language processing. *Bull Med Libr Assoc*, 81(2):184–194, 1993.
72. A.T. McCray, O. Bodenreider, J.D. Malley, and A.C. Browne. Evaluating umls strings for natural language processing. In *Proc. AMIA Symp. 2001*, pages 448–452, 2001.
73. A.T. McCray, A.C. Browne, and O. Bodenreider. The lexical properties of the gene ontology. In *Proc. AMIA Symp. 2002*, pages 504–508, 2002.
74. D.J. McGoldrick and L. Hunter. Descriptron: A web service for information management in the medical and biological sciences. in preparation.
75. MedMiner. `http://discover.nci.nih.gov/textmining/main.jsp`.
76. MetaMap. `http://mmtx.nlm.nih.gov/`.
77. A. Morgan, L. Hirschman, A. Yeh, and M. Colosimo. Gene name extraction using flybase resources. In *Proceedings of the ACL 2003 workshop on natural language processing in biomedicine*. Association for Computational Linguistics, 2003.
78. MUC. Message understanding conference proceedings. `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/proceedings_index.html`.

79. M. Narayanaswamy, K.E. Ravikumar, and K. Vijay-Shanker. A biological named entity recognizer. In *Pacific Symposium on Biocomputing 8*, pages 427–438, 2003.
80. P.V. Ogren, K.B. Cohen, G.K. Acquaah-Mensah, J. Eberlein, and L. Hunter. The compositional structure of gene ontology terms. In *Proc. of the Pacific Symposium on Biocomputing 2004*, pages 214–225, 2004.
81. T. Ohta, Y. Tateisi, J.-D. Kim, H. Mima, and J.-I. Tsujii. The genia corpus: an annotated corpus in molecular biology. In *Proceedings of the Human Language Technology Conference*, 2002.
82. D.E. Oliver, G. Bhalotia, A.S. Schwartz, R.B. Altman, and M.A. Hearst. Tools for loading medline into a local relational database. submitted.
83. T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.
84. Porter. Porter stemmer site. `http://www.tartarus.org/~martin/PorterStemmer/`.
85. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
86. S. Raychaudhuri, J.T. Chang, P.D. Sutphin, and R.B. Altman. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Research*, 12:203–214, 2002.
87. C.K. Riesbeck. Conceptual analyzer to direct memory access parsing: an overview. In N.E. Sharkey, editor, *Advances in cognitive science I.* Ellis Horwood Ltd, 1986.
88. E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proc. of the eleventh national conference on artificial intelligence (AAAI-93)*, pages 811–816. AAAI Press/MIT Press, 1993.
89. T.C. Rindflesch and A.R. Aronson. resolution while mapping free text to the umls metathesaurus. In *Proc Annu Symp Comput Appl Med Care 1994*, pages 240–244, 1994.
90. T.C. Rindflesch, L. Tanabe, J.N. Weinstein, and L. Hunter. Edgar: extraction of drugs, genes, and relations from the biomedical literature. In *Pacific Symposium on Biocomputing 5*, pages 514–525, 2000.
91. W. Ruan, H. Long, D.H. Vuong, and Y. Rao. Bifocal is a downstream target of the ste20-like serine/threonine kinase misshapen in regulating photoreceptor growth cone targeting in drosophila. *Neuron*, 36(5):831–842, 2002.
92. G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Publishing Company, 1989.
93. R.C. Schank and R.P. Abelson. *Scripts, plans, goals, and understanding.* Halsted Press, 1976.
94. Schwartz and Hearst. Schwartz and hearst abbreviation code site. `http://biotext.berkeley.edu/software.html`.
95. A.S. Schwartz and M.A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, pages 8:451–462, 2003.
96. D.B. Searls. The computational linguistics of biological sequences. In L. Hunter, editor, *Artificial Intelligence and Molecular Biology*, pages 47–121. MIT Press, 1993.
97. D.B. Searls. The language of genes. *Nature*, 420:211–217, 2002.

98. H. Shatkay, S. Edwards, W.J. Wilbur, and M. Boguski. Genes, themes and microarrays: using information retrieval for large-scale gene analysis. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology 2000*, pages 317–328, 2000.
99. H. Shatkay and R. Feldman. Mining the biomedical literature in the genomic era: an overview. *Journal of Computational Biology*, 10(6):821–855, 2004.
100. M. Skounakis and M. Craven. Evidence combination in biomedical natural-language processing. In *Third workshop on data mining in bioinformatics*, 2003.
101. M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov models for information extraction. In Morgan Kaufmann, editor, *Proceedings of the 18th international joint conference on artificial intelligence*, 2003.
102. Slackware.
103. G.W. Smith. *Computers and human language*. Oxford University Press, 1991.
104. Stanford. Stanford abbreviation server. `http://bionlp.stanford.edu/abbreviation/`.
105. P.D. Stetson, S.B. Johnson, M. Scotch, and G. Hripcsak. The sublanguage of cross-coverage. In *Proceedings of the AMIA 2002 Annual Symposium*, pages 742–746, 2002.
106. L. Tanabe. *Text mining the biomedical literature for genetic knowledge.* PhD thesis, George Mason University doctoral dissertation., 2003.
107. L. Tanabe, U. Scherf, L.H. Smith, J.K. Lee, G.S. Michaels, and L. Hunter et al.
108. L. Tanabe and W.J. Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002a.
109. L. Tanabe and W.J. Wilbur. Tagging gene and protein names in full text articles. In *Proc. of the workshop on natural language processing in the biomedical domain.* Association for Computational Linguistics, 2002b.
110. ThemeRiver. Themeriver demo. `http://www.pnl.gov/infoviz/technologies.html#themeriver`.
111. TrigramsAndTags. Trigrams'n'tags (tnt) site. `http://www.coli.uni-sb.de/~thorsten/tnt/`.
112. UMLS. `http://www.nlm.nih.gov/research/umls/`.
113. C. Verspoor, C. Joslyn, and G.J. Papcun. The gene ontology as a source of lexical semantic knowledge for a biological natural language processing application. In *Participant Notebook of the ACM SIGIR'03 Workshop on Text Analysis and Search for Bioinformatics*, pages 51–56, 2003.
114. H.M. Wain, M. Lush, F. Ducluzeau, and S. Povey. Genew: the human gene nomenclature database. *Nucleic Acids Research*, 30(1):169–171, 2002.
115. A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii. Event extraction from biomedical papers using a full parser. In *Pacific Symposium on Biocomputing 2001*, pages 408–419, 2001.
116. M.D. Yandell and W.H. Majoros. Genomics and natural language processing. *Nature Reviews/Genetics*, Vol. 3:601–610, Aug. 2002.
117. Yapex. Yapex data set. `http://www.sics.se/humle/projects/prothalt/#data`.
118. Yapex. Yapex entity identification system site. `http://www.sics.se/humle/projects/prothalt/`.
119. A. Yeh, L. Hirschman, and A. Morgan. Evaluation of text data mining for database curation: lessons learned from the kdd challenge cup. *Bioinformatics*, 19(Suppl. 1):i331–i339, 2003.

120. H. Yu, C. Friedman, A. Rhzetsky, and P. Kra. Representing genomic knowledge in the umls semantic network. In *Proc AMIA Symp 1999*, pages 181–185, 1999.