Chapter 18

GENE PATHWAY TEXT MINING AND VISUALIZATION

Daniel M. McDonald[1], Hua Su[1], Jennifer Xu[1], Chun-Ju Tseng[1], Hsinchun Chen[1], and Gondy Leroy[2]

[1]University of Arizona, Management Information Systems Department, Tucson, AZ 85721;
[2]Claremont Graduate University, School of Information Science, Claremont, CA 91711

Chapter Overview

Automatically extracting gene-pathway relations from medical research texts gives researchers access to the latest findings in a structured format. Such relations must be precise to be useful. We present two case studies of approaches used to automatically extract gene-pathway relations from text. Each technique has performed at or near the 90 percent precision level making them good candidates to perform the extraction task. In addition, we present a visualization system that uses XML to interface with the extracted gene-pathway relations. The user-selected relations are automatically presented in a network display, inspired by the pathway maps created by gene researchers manually. Future research involves identification of equivalent relations expressed differently by authors and identification of relations that contradict each other along with the inquiry of how this information is useful to researchers.

Keywords

text mining; visualization; gene pathway; PubMed; MEDLINE; information extraction; linguistic analysis; co-occurrence

1. INTRODUCTION

The PubMed database is a valuable source of biomedical research findings. The collection contains information for over 12 million articles and continues to grow at a rate of 2,000 articles per week. The rapid introduction of new research makes staying up-to-date a serious challenge. In addition, because the abstracts are in natural language, significant findings are more difficult to automatically extract than findings that appear in databases such as SwisProt, InterPro, and GenBank. To help alleviate this problem, several tools have been developed and tested for their ability to extract biomedical findings, represented by semantic relations from PubMed or other biomedical research texts. Such tools have the potential to assist researchers in processing useful information, formulating biological models, and developing new hypotheses. The success of such tools, however, relies on the accuracy of the relations extracted from text and utility of the visualizers that display such relations. We will review existing techniques for generating biomedical relations and the tools used to visualize the relations. We then present two case studies of relation extraction tools, the Arizona Relation Parser and the Genescene Parser along with an example of a visualizer used to display results. Finally, we conclude with discussion and observations.

2. LITERATURE REVIEW/OVERVIEW

In this section we review some of the current contributions in the field of gene-pathway text mining and visualization.

2.1 Text Mining

Published systems that extract biomedical relations vary in the amount and type of syntax and semantic information they utilize. Syntax information for our purposes consists of part-of-speech (POS) tags and/or other information described in a syntax theory, such as Combinatory Categorical Grammars (CCG) or Government and Binding Theory. Syntactic information is usually incorporated via a parser that creates a syntactic tree. Semantic information, however, consists of specific domain words and patterns. Semantic information is usually incorporated via a template or frame that includes slots for certain words or entities. The focus of this review is on the syntax and semantic information used by various published approaches. We first review systems that use predominantly either syntax or semantic information in relation parsing. We then review systems that more

equally utilize both syntax and semantic information via pipelined analysis. In our review, we will also draw connections between the amount of syntax and semantic information used and the size and diversity of the evaluation.

2.1.1 Syntax Parsing

Tools that use syntax parsing seek to relate semantically relevant phrases via the syntactic structure of the sentence. In this sense, syntax serves as a "bridge" to semantics (Buchholz, 2002). However, syntax parsers have reported problems of poor grammar coverage and over generation of candidate sentence parses. In addition, problems arise because important semantic elements are sometimes widely distributed across the parse of a sentence and parses often contain many syntactically motivated components that serve no semantic function (Jurafsky and Martin, 2000). To handle these challenges, some filtering is performed to eliminate non-relevant parses. Also, sentence relevance is judged before parsing to avoid parsing irrelevant sentences. As reported in the literature, however, systems relying primarily on syntax parsing generally achieve lower precision numbers as compared to relations extracted from systems using full semantic templates.

In the following predominantly syntactic systems, key substances or verbs are used to identify relevant sentences to parse. Park et al. used a combinatory categorical grammar to syntactically parse complete sentence structure around occurrences of proteins (Park et al., 2001). Sekimizu et al. used partial parsing techniques to identify simple grammatical verb relations involving seven different verbs (Sekimisu et al., 1998). Yakushiji et al. used full syntax parsing techniques to identify not just relations between substances, but the sequence of the relations as they occurred in events (Yakushiji et al., 2001). Others, while still predominantly syntactic, have incorporated different types of semantic information. Leroy et al. used shallow syntax parsing around three key prepositions to locate relevant relations (Leroy et al., 2003). Thomas et al. reported on their system Highlight that used partial parsing techniques to recognize certain syntactic structures (Thomas et al., 2000). Semantic analysis was then incorporated afterwards by requiring certain syntactic slots to contain a certain type of semantic entity. In addition, the system extracted only relations that used one of the verb phrases *interact with*, *associate with*, or *bind to*.

With the exception of Leroy et al. that reported a 90 percent precision, the highest precision reported from the syntax approaches did not exceed 83 percent. Park et al. reported 80 percent precision. Sekimizu et al. reported 83 percent precision. Yakushiji et al. reported a recall of 47 percent. The Highlight system reported a high of 77 percent precision. Semantic

approaches on the other hand have achieved precision as high as 91 and 96 percent (Friedman et al., 2001; Pustejovsky, Castano et al., 2002).

Despite the lower performance numbers, the evaluations of syntax approaches typically involved more documents than used in semantic parser evaluations. Park et al. evaluated their parser on 492 sentences, while Leroy et al. used 26 abstracts, and Thomas et al. used 2,565 abstracts. Semantic parsers have been evaluated on single articles or more heavily constrained to extract only inhibit relations. Using a greater number of documents in an evaluation requires a parser to deal with more varied writing styles and topic content and thus performance tends to be lower.

2.1.2 Semantic Templates

Other systems rely more on semantic information than on syntax. Semantic parsing techniques are designed to directly analyze the content of a document. Rules from semantic grammars correspond to the entities and relations identified in the domain. Semantic rules connect the relevant entities together in domain-specific ways. Rindflesch et al. incorporate a greater amount of semantic information in their system EDGAR (Rindflesch et al. 2000). Documents are first shallow parsed and important entities are identified using the Unified Medical Language System (UMLS). Biomedical terms are then related together using semantic and pragmatic information. Performance was described as “moderate”. GENIES (Friedman et al. 2001) and a system reported by Hafner and colleagues (Hafner et al., 1994) rely primarily on a semantic grammar. GENIES starts by recognizing genes and proteins in journal articles using a term tagger. The terms are then combined in relations using a semantic grammar with both syntactic and semantic constraints. The system was tested on one journal article with the reported precision of 96 percent and a recall of 63 percent. In the system developed by Hafner and colleagues, a semantic grammar was developed to handle sentences with the verbs *measure*, *determine*, *compute*, and *estimate*. The grammar contained sample phrases acceptable for the defined relations. The system was in an early state of development when reported. Pustejovsky et al. used a semantic automaton that focused on certain verbal and nominal forms. Precision of 91 percent was reported along with a recall of 59 percent. The evaluation, however, only extracted relations that used the verb *inhibit*.

Semantic approaches, while more precise, are subject to poorer coverage than syntax approaches. As a result, semantic systems are often evaluated using a smaller sample of documents or a smaller sample of relevant sentences. GENIES was evaluated using one full text article. Pustejovsky et

al. limited their relations of interest to *inhibit* relations, and Hafner et al. and Rindflesch et al. did not submit precision or recall numbers.

2.1.3 Balanced Approaches

Balanced approaches utilize more equal amounts of syntax and semantic processing. Syntax processing takes place first, often resulting in an ambiguous parse. More than 100 parses can be generated for a single sentence (Novichkova, 2003). Semantic analysis is then applied to eliminate the incorrect syntactic parse trees and further identify domain words such as proteins and genes. In this fashion, systems combine the flexibility of syntax parsing with the precision of semantic analysis. Such combination has resulted in systems that have been evaluated over a large numbers of documents. Despite the use of both syntactic and semantic processing, however, problems specific to syntactic and semantic analysis persist in part because the analyses are still separate. Syntax grammars remain subject to poor coverage. Because semantic analysis only occurs after syntactic processing, a syntax grammar with poor coverage cannot be improved by the semantic analysis. At the same time, a syntax grammar with good coverage can still generate more parses than can be effectively disambiguated using semantic analysis.

Gaizauskas et al. reported on PASTA, a system that included complete syntax and semantic modules (Gaizauskas et al., 2003). The relation extraction component of PASTA was evaluated using 30 unseen abstracts. Recall was reported at 68 percent, among the highest recall number published, and a precision of 65 percent. The high recall and larger number of documents in the experiment suggest relatively good coverage of their syntax grammar. The relatively lower precision number reflects the sparser coverage of the semantic module given the incoming syntactic parses and their task of extracting protein interactions.

Novichkova et al. reported on their system MedScan which involved both syntax and semantic components (Novichkova et al., 2003). Their first evaluation focused on the coverage of their syntax module, which was tested on 4.6 million relevant sentences from PubMed. Their syntax grammar produced parses for 1.56 million sentences out of the 4.6 million tested resulting in 34 percent coverage. In a more recent study, Daraselia et al. reported a precision of 91 percent and a recall of 21 percent when extracting human protein interactions from PUBMED using MedScan (Daraselia et al., 2004). Such a high precision supports the robustness of their semantic analysis given their task. However, the recall of 21 percent still shows the problem of a syntax grammar with relatively poor coverage. Balancing the use of syntax and semantic analysis contributed to MedScan's ability to

perform well on a large sample size. Adding semantic analysis to the pipe, however, did not improve the coverage of the syntax grammar.

2.2 Visualization

Visualization plays an important role in helping researchers explore, comprehend, and analyze gene pathways. Gene pathways often take the form of a graph, in which nodes represent gene and gene products such as proteins and RNAs and edges represent interactions and reactions. Many gene databases and analytical packages provide pathway visualization functionality. KEGG Kanehisa and Goto, 2000; Kanehisa et al., 2002), for instance, provides manually drawn graph representations of protein interaction networks to help users understand the functions of the cell or an organism. However, because manually drawn pathways reflect only the knowledge at the time of drawing and often are difficult to query and update, many analytical packages have employed automatic techniques to visualize gene pathway graphs.

Automatic graph visualization is a well-studied topic in the information visualization area. Graph layout, visual cues, navigation, and interactivity are the major issues associated with graph visualization. Different pathway analysis packages employ different approaches to address these issues.

- *Graph layout*. The key of graph layout is to calculate the position of each node for a given graph according to certain aesthetic criteria or constraints (Herman et al., 2000). Aesthetically pleasing graphs may be generated if nodes and edges are distributed evenly and the number of crossing edges are kept to a minimum (Purchase, 2000). Layout techniques, such as tree layout, circular layout, and spring models have been employed in several gene pathway packages. The tree layout arranges a graph in a hierarchical structure by positioning children nodes beneath parent nodes. In the circular layout, all nodes are placed on a circle and edges are drawn between these nodes. The spring algorithm models a graph as a force-directed energy system with parcels connected by springs. The nodes attract and repulse one another and the system settles down when the energy of the system is minimized (Eades, 1984). G.NET, for example, presents gene pathways using both tree layout and a spring model. Osprey (Breitkreutz et al., 2003) and the Pathway Tools (Karp et al., 2002) offer multiple layout options including linear, tree, and circular to present gene pathways.
- *Visual cues*. Because data about components in gene pathways often are of multiple dimensions, various visual cues have been used to represent different attributes of gene pathway components. For example, in GeneNet (Ananko et al., 2002) the multimerisation state of a protein is

represented by the shape of the protein icon, while the functional state of a protein is represented by colors. Another example is Osprey (Breitkreutz et al., 2003) in which both gene and gene interactions are color-coded to indicate the biological process of the gene, or the experimental system and data source of an interaction.

- *Navigation and interactivity*. As the size of gene pathways increases, the graph may become fairly cluttered and difficult to read. The most commonly used method to address this problem is the zooming operation (Herman et al., 2000). By zooming into a selected area in a graph a user can focus on its details. Most existing pathway systems allow zooming in graphs. However, a user may be disoriented while navigating and zooming between different areas of a graph because of the loss of contextual information. One solution to the focus+context problem is to present the local view and global view of a graph simultaneously. In the G.Net system, the main window presents the selected local area of a pathway and the global view of the graph is shown in a smaller window. The user can select a focal area in the global view and the main window shows the details of the focal area. Another solution is the fisheye view (Furnas, 1986) and hyperbolic tree (Lamping and Rao, 1986). Although on the basis of different mechanisms, they both maintain the focus and context by enlarging the focal area and displaying other parts of a graph with less detail to help user navigate on a complex graph. Fisheye view technology is employed in the GSCope system to visualize highly complicated biomolecular networks (Toyoda et al., 2003).

In addition to these fundamental issues associated with graphs, graph query, editing, filtering, and pathway comparison are also addressed in many pathway analysis systems. For example, Osprey (Breitkreutz et al., 2003) and GeneNet (Ananko et al., 2002) allow users to search for specific genes and filter out unneeded information based on data source, function, etc. BioMiner (Sirava et al., 2002) and ToPLign (Kuffner et al., 2000) support the comparison of metabolic pathways among different organisms.

3. CASE STUDIES/EXAMPLES

We now present three case studies, which include two implementations of parsers that extract gene-pathway information and one system for visualizing the pathways extracted.

3.1 Arizona Relation Parser

In the Arizona Relation Parser (ARP), syntax and semantic analysis are applied together in one parsing process as opposed to the pipelined approach that applies syntax and then semantic analysis in sequence. Others have shown such a combination to be effective for information extraction (Ciravegna and Lavelli,, 1999), but we have not seen such a combination in the biomedical domain. We propose that the benefits of combining syntax and semantic analysis can be realized by using a greater number of word classes or tags that reflect the relevant properties of words. Constraints limiting the type of combinations that occur are thus implicit by the absence of such parsing rules. With a greater number of tags, parsing rules must be explicitly written for each word class. When a rule is created and added to the system, it may be correct based on syntax, semantics, or some combination. The theory behind the rules is only implicit. While many rules have to be written to support the numerous tags, semantic constraints do not have to be specified in the system's lexicon. Such an approach differs from most semantic parsing in that we attempt to parse the entire sentence into relations, regardless of the verbs used. Even if a triple is not relevant to the task, ARP still extracts it. Semantic parsing approaches use templates that center around and are specific to key verbs. Such parsing approaches do not parse non-relevant structures. Semantic approaches are thus more tailored to a specific domain and require relations to be anticipated to be extracted. Our hybrid approach can extract a relation that has not been so highly specified, but requires an accurate filtering mechanism to remove non-relevant relations.

ARP combines syntax and semantic analysis together by introducing over 150 new word classes to separate words with different properties. In comparison, the PENN TREE BANK has approximately 36 common word classes. The majority of the new word classes are semantically or lexically oriented, while we also carry over a subset of the syntax tags from the PENN TREE BANK tag set. A sample of the tags is shown in Table 18-1. The set of tags was chosen using three primary methods. First, we started with a complete lexicon extracted from the PENN TREE BANK and BROWN corpora. Then the most common prepositions and verbs from our 40-abstract training set that had been assigned multiple part-of-speech tags from the PENN TREE BANK lexicon were then assigned a unique tag in our lexicon. Second, domain relevant nouns were sub classed into groups of relevant substances or entities. Third, many of the common 36 PENN TREE BANK syntax tags were included in the new tag set. The role of new tags is determined by the way they can be parsed. As tags take semantic and syntactic properties, rules that apply to the tags reflect semantic and

syntactic phenomena. Using over 150 new tags with a regular grammar eliminates the problem of over generating parses. Two different parsing rules are not allowed to act on the same input token sequence. Only one parse tree is generated for each sentence, with only two levels being analyzed for relation extraction. The particulars of the parsing process now follow. Output of the parsing steps is shown in Figure 18-1 below. The boxed numbers on the left in the figure correspond to the subsection numbers listed beside the text below.

Table 18-1. A sample of tags used by the Arizona Relation Parser

Selection Method	Tags
Unique tags in our lexicon obtained by observing ambiguous tags from the PENN TREE BANK	BE, GET, DO, KEEP, MAKE, INCD (include), COV (cover), HAVE, INF (infinitive), ABT (about), ABOV, ACROS, AFT (after), AGNST, AL (although), AMG (among), ARD (around), AS, AT, BEC, BEF (before) , BEL (below), BTN (between), DUR (during), TO, OF, ON, OPP (neg/opposite), OVR (over), UNT (until), UPN (upon), VI (via), WAS (whereas), WHL (while), WI (with), WOT
Domain relevant noun classes	DATE, PRCT, TIME, GENE, LOCATION, PERSON, ORGANIZATION
PENN TREE BANK syntax tags	IN, NP, VBD, VBN, VBG, VP, NN, NNP, NNS, NNPS, PRP, PRP$, RB, RBR

3.1.1 Sentence and Word Boundaries

The parsing process begins with tokenization, where word and sentence boundaries are recognized. The sentence splitting relies on a lexicon of 210 common abbreviations and rules to recognize new abbreviations. Documents are tokenized generally according to the PENN TREE BANK tokenizing rules. In addition, words are also split on hyphens, a practice commonly performed in the bioinformatics domain (Gaizauskas et al., 2003). A phrase tagger based on a finite state automaton (FSA) is also run during this step to recognize words that are best tokenized as phrases. Common idiomatic and discourse phrases (i.e. “for example” and “on the other hand”) are grouped together in this step along with other compound lexemes, such as compound gene names. Such phrases receive a single initial tag, despite being made up of multiple words.

3.1.2 Arizona Part-of-Speech (POS)/Semantic Tagger

We developed a Brill-style tagger (Brill, 1993) written in Java and trained on the Brown and Wall Street Journal corpora. The tagger was also trained using 100 PubMed abstracts and its lexicon was augmented by the

words and tags from the GENIA corpus (Ohta et al., 2002). The tags used to mark tokens include the 150 new tags (generated by methods described earlier) along with the original tags from the PENN TREE BANK tag set.

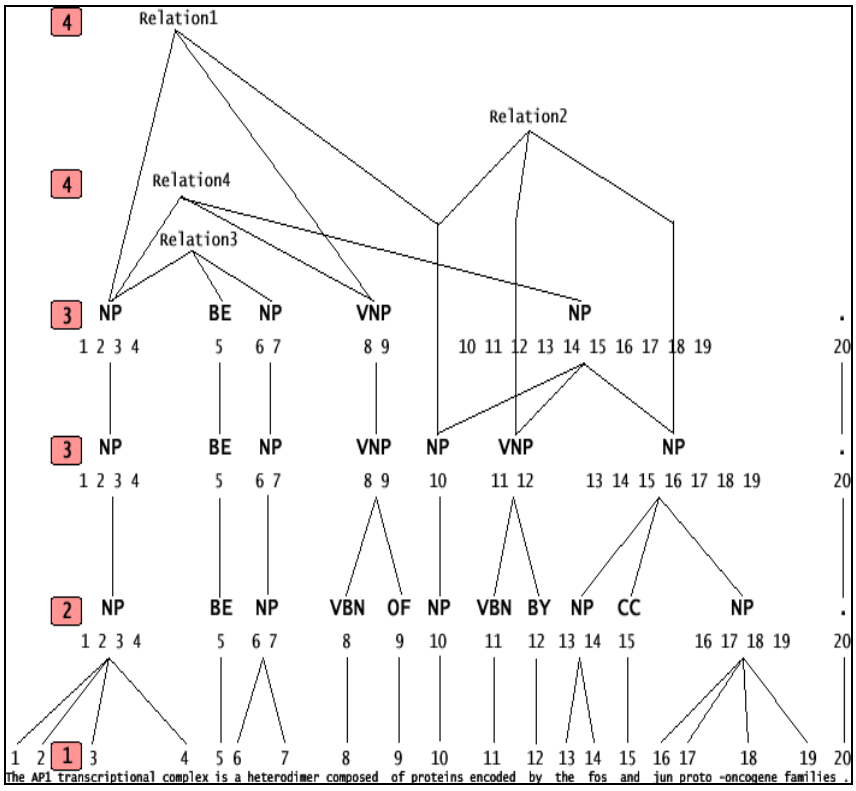

Figure 18-1. An example of a hybrid parse tree.

3.1.3 Hybrid Parsing

In this step word and phrases are combined into larger phrase classes consistent with the role the phrases play in the sentence. The parser output appears in boxed-number 3 shown in Figure 18-1. We attempt to address the poor grammar coverage problem by relaxing some of the parsing assumptions made by full-sentence parsers. For example, full sentence parsing up to a root node of a binary branching tree is not required because we extract semantic triples. Thus, ARP uses a shallow parse structure with n-ary branching, such that any number of tokens up to 24 can be combined into a new node on the tree. Internally, a cascade of five finite state automata

attempts to match adjacent nodes from the parse chart to rules found in the grammar. Each FSA handles specific grammatical constructs:

- Level 0: Pre-parsing step where simple noun and verb groups are recognized.
- Level 1: Conjunctions are recognized and combined as noun phrases or treated as discourse units.
- Level 2: Prepositions are attached to verb phrases where possible and made into prepositional phrases elsewhere.
- Level 3: This level catches parsing that should have taken place at level 1 or 2, but did not because of embedded clauses or other preprocessing requirement.
- Level 4: Relative and subordinate clauses are recognized.

The ARP utilizes a regular grammar that handles dependencies up to 24 tags or phrase tags away. Most sentences end before reaching this limit. Regular grammars have previously been used to model the context-sensitive nature of the English language, notably in FASTUS and its medical counterpart HIGHLIGHT (Hobbs et al., 1996). Different in our approach, however, is that the grammar rules are constrained by surrounding tags and thus are only fired when rule core and rule context tags are filled. Figure 18-2 gives an example of a grammar rule, shown on the left, with several abbreviated rules shown on the right. The rule pattern in Figure 18-2 consists of the string “BY NP CC NP .”, the rule core equal to “NP CC NP”. The rule core is transformed to a “NP” when the entire rule pattern is matched. Therefore, the rule core has to be preceded by a “BY” tag and be followed by a “.” tag to be combined into a new noun phrase. The GRAMMAR LEVEL designation, in Figure 18-2, refers to which cascade of the five uses this rule.

```
<GRAMMAR LEVEL= “1”>
<RULE NUM=1>
<RULEPATTERN>
<PREVIOUSCONTEXT TAG=”BY” />
<RULECORE>NP CC NP</RULECORE>
<FUTURECONTEXT TAG=”.” />
</RULEPATTERN>
<TRANSFORMATION>
NP
</TRANSFORMATION>
</RULE>
</GRAMMAR>
```

Figure 18-2. A single parsing rule with a rule pattern and transformation

Figure 18-3 below shows an abbreviated notation for grammar rules, showing nine in total. The rule core is in bold and the rule context is italicized. Some of the rules listed have empty rule context slots.

INP VDP NP . transforms to>>INP
IT VP **NP CC NP** *:* transforms to>>NP
AFT NP transforms to>>WHENP
AGNST NP transforms to>>AGPP
AMG NP CC NP II transforms to>>AMGP
ARD NP , transforms to>>ARDP
BE **NP VDP NP** transforms to>>NP
BE **RB JJ** transforms to>>JJ
WI NP CC NP , transforms to>>MOD

Figure 18-3. Abbreviated notation for grammar rules

3.1.4 Relation Identification

The top two levels of the parse chart are passed to the relation identification step. Relations can be loosely compared to subject, verb, object constructs and are extracted using knowledge patterns. Knowledge patterns refer to the different syntactic/semantic patterns used by authors to convey knowledge. Like the parsing rules, knowledge patterns consist of rule patterns with rule context and rule core and transformations that are applied only on the rule core. Different from the parsing rules, however, are the actions that take place on the rule core. First, the rule core does not get transformed into a single new tag, but rather each tag in the rule core is assigned a role, from a finite set of roles *R*. Currently, there are 10 different roles defined in the set *R*. Roles 0 - 3 account for the more directly expressed knowledge patterns. Roles 4 - 9 identify nominalizations and relations in agentive form. Sentences may contain multiple overlapping knowledge patterns, with tags playing multiple roles. The relation parsing output is shown in Figure 18-1 next to the boxed number 4. Conjunctions in the relations are split after the relation extraction step.

Once potential relations are identified, each relation has to meet a number of semantic constraints in order to be extracted. In the current system, at least one word from the first argument and at least one word from the second argument had to exist in a gene/gene products lexicon, such as those from the Gene Ontology and HUGO. In addition, relations were limited by filtering predicates with 147 verb stems. At least one of the words in the predicate had to contain a verb stem. Examples of verb stems from the lexicon include *activat, inhibit, increas, suppress, bind, catalyz, block,*

augment, elicit, promot, revers, control, coregulat, encod, downregulat, destabiliz, express, hydrolye, inactivat, interfer, interact, mimic, neutraliz, phosphorylat, repress, trigger, and induc. Our biology expert generated the list of relevant verb stems by examining verbs appearing in PubMed.

3.1.5 RESULTS

The performance of the hybrid grammar together with the semantic filtering was tested in an experiment involving 100 unseen abstracts. Fifty abstracts were used to test precision and 50 for recall. The 100 abstracts were randomly selected from a collection of 23,000 abstracts related to the AP-1 family of transcription factors extracted from PubMed. For the precision experiment, an expert with a Ph.D. in biology separated the parser-generated relations into four different categories. Typically, substance-only relationships are extracted from PubMed texts. Substance-only relationships represent our category A and category B relations. Category A relations were genetic regulatory pathway relations between two recognizable substances. An example of a category A relation is "inhibit(MDM2, SMAD3)." Category B relations were gene-pathway relations between at least one substance and a process or function. An example of a category B relation is "regulates(counterbalance of protein-tyrosine kinases, activation of T lymphocytes to produce cytokines)." This group of relations contained the substances that appeared in gene-pathway maps. In addition to substance relations, we had our expert identify which of the non-substance relations would be "relevant" for pathway map creation. This type of relations, termed category C, is a more open-ended class of relations that is not typically measured in evaluations. Category C relations consisted of more general biologically relevant relations, an example being "mediate(target genes, effects of AP-1 proteins)." ARP should more effectively extract category C relations than purely semantic approaches because it parses every sentence, instead of just sentences with certain verbs or semantic constructions. Category D relations were incorrect or only partially relevant.

The primary results from the experiment are listed in Table 18-2. The parser extracted 130 relations from 50 abstracts, with 79 of those relations belonging to categories A or B. Thus the precision of the parser in extracting pathway relations was 61 percent. When we widened the pool of correct relations to include pathway-relevant relations (category C), correct relations

Table 18-2. Parser performance results

Precision (categories A & B) after filtering	79/130	61%
Recall (categories A & B) after filtering	43/125	35%

jumped to 116 producing an 89 percent precision number, as shown in Table 18-3. The ability to capture category C relations, which are more open-ended, shows the strength of the hybrid approach. By adding category C to the experiment, we affirm the parsing component is performing well, while the semantic filtering function lacks precision. Our current filtering approach did not distinguish well between the A/B group and the C group.

Table 18-3. Parser performance less filtering errors

Precision (A, B & C) after filtering	116/130	89%
Recall (categories A & B) before filtering	76/125	61%

To perform the recall experiment, an expert in biology manually identified all gene pathway relations from categories A and B from 50 randomly selected unseen abstracts. She identified a total of 125 pathway relations between substances from 36 of the 50 abstracts. Fourteen of the abstracts produced no relations. Recall equaled the ratio of system-identified relations to the expert identified relations. Table 18-2 shows the system's recall score of 35 percent. In addition to the standard recall score, we wanted to show the recall of the parsing component unaffected by semantic filtering. Table 18-3 shows the parser extracted 61 percent of the relations.

Including the errors introduced by the semantic filtering component, Table 18-4 lists the reasons why relations were not extracted. The largest number of relations was missed due to imprecise filtering. An incomplete lexicon caused the majority of filtering errors. We are in the process of replacing our semantic filtering step with a biological named entity extraction module to overcome this problem. The next largest group of relations was missed due to incomplete extraction rules. Since this evaluation, the number of extraction rules has more than doubled (rules totaled 210 at the time of the experiment). The third largest group of relations was missed due to the parser's inability to handle co-reference. The expert identified relations that required co-reference resolution 12.5 percent of the time. The co-reference usually occurred between sentences. Along with the entity identification module, a co-reference module is being developed to address this problem. Finally, the parser missed 2.7 percent of the expert identified relations due to parsing errors.

Table 18-4. Why the parser did not recall relations

Reason	% Missed
Removed at semantic filtering stage	26.3%
Incomplete extraction rules	23.6%
Required co-reference information	12.5%
Parsing error	2.7%

3.2 GeneScene Parser

This case study discusses the Genescene parser, which extracts relations between noun phrases and is tuned for biomedical text.

3.2.1 Extracting Semantic Elements

The parser begins by formatting, tokenizing, and tagging the PUBMED abstract with part-of-speech and noun phrase tags. The abstracts are prepared by removing phrases referring to publisher and copyright information. Then the sentence splitter is run followed by the AZ Noun Phraser (Tolle and Chen, 2000) to extract noun phrases. Verbs and adverbs are tagged with their part-of-speech (POS) based on a rule set and lexical lookup in the UMLS Specialist Lexicon. Closed class words such as prepositions, negation, conjunctions, and punctuation are also tagged. Nouns and noun phrases are checked for nominalizations. When a nominalization is discovered, e.g., "activation," then both the infinitive and the original nominalization are retained. Nominalizations can be replaced by the infinitive to facilitate text mining and visualization.

3.2.2 Extracting Structural Elements

Relations have a syntactic basis: they are built around basic sentence structures and prepositions. Prepositions were chosen because they form a closed class and can help capture the structure of a sentence. The closed classes' membership does not change and allows us to build very specific but semantically generic relation templates. In addition, prepositions often head phrases (Pullum and Huddleston, 2002) and indicate different types of relations, such as time or spatial relations (Manning and Schütze, 2001). Although prepositional attachment ambiguity may become a problem, we believe that researchers in biomedicine use a common writing style and so the attachment structures will not vary much for a specific structure.

This case study describes relations built around three prepositions, "by," "of," and "in," which occur frequently in text and lead to interesting, diverse biomedical relations. "By" is often used to head complements in passive sentences, for example in "Mdm2 is not increased by the Ala20 mutation." "Of" is one of the most highly grammaticised prepositions (Pullum and Huddleston, 2002) and is often used as a complement, such as for example in "the inhibition of the activity of the tumor suppressor protein p53." "In" is usually an indication of location. It forms interesting relations when combined with verbs, for example in "Bcl-2 expression is inhibited in precancerous B cells."

Negation is captured as part of these relations when there are specific nonaffixal negation words present. This is the case for *Not*-negation, e.g., not, and *No*-negation, e.g., never. We do not deal with inherent negatives (Tottie, 991), e.g., deny, which have a negative meaning but a positive form. Neither do we deal with affixal negation. These are words ending in –less, e.g., childless, or starting with non–, e.g., noncommittal.

3.2.3 Combining Semantic and Structural Elements

Cascaded, deterministic finite state automata (FSA) describe the relations: basic sentences (BS-FSA) and relations around the three prepositions (OF-FSA, BY-FSA, IN-FSA). All extracted relations are stored in a database. They can contain up to 5 elements but require minimally 2 elements. The *left-hand side* (*LHS*) of a relation is often the active component and the *right-hand side* (*RHS*) the receiving component. The *connector* connects the LHS with the RHS and is a verb, preposition, or verb-preposition combination. The relation can also be *negated* or augmented with a *modifier*. For example from "Thus hsp90 does not inhibit receptor function solely by steric interference; rather …" the following relation is extracted "NOT(negation): Hsp90 (LHS) – inhibit (connector) – receptor function (RHS)." Passive relations based on "by" are stored in active format. In some cases the connector is a preposition, e.g., the relation "single cell clone – of – AK-5 cells." In other cases, the preposition "in" and the verb are combined, e.g., the relation "NOT: RNA Expression – detect in – small intestine".

The parser also recognizes coordinating conjunctions with "and" and "or." A conjunction is extracted when the POS and a UMLS Semantic Types of the constituents fit. Duplicate relations are stored for each constituent. For example, from the sentence "Immunohistochemical stains included Ber-EP4, PCNA, Ki-67, Bcl-2, p53, SM-Actin, CD31, factor XIIIa, KP-1, and CD34," ten relations were extracted based on the same underlying pattern: "Immunohistochemical stains - include - Ber-EP4," "Immunohistochemical stains - include - PCNA," etc.

3.2.4 Genescene Parser Evaluation

1. Overall Results

In a first study, three cancer researchers from the Arizona Cancer Center submitted 26 abstracts of interest to them. Each evaluated the relations from his or her abstracts. A relation was only considered correct if each component was correct and the combined relation represented the

information correctly. On average, there were 13 relations extracted per abstract and 90 percent were correct.

2. FSA-specific Results

We also report whether the relations were extracted by the appropriate FSA and calculate precision and recall (see Table 18-5) and coverage. All details of this study can be found in (Leroy et al., 2003).

Table 18-5. Genescene: Precision and Recall

FSA	Total Correct	Total Extracted	Total in Text	Precision (%)	Recall (%)
Relationships:					
BS-FSA:	8	15	23	53	35
OF-FSA:	145	157	203	92	71
BY-FSA:	15	17	24	88	63
IN-FSA:	11	13	37	85	30
All:	**179**	**202**	**287**	**89**	**62**
Conjunctions:					
BS-FSA:	1	1	1	100	100
OF-FSA:	10	10	22	100	45
BY-FSA:	0	0	1	-	0
IN-FSA:	1	1	6	100	16
All:	**12**	**12**	**30**	**100**	**40**

Precision was calculated by dividing the number of correct relations by the total number of extracted relations. The correct relations are those relations considered correct by the researchers, as described above, but with the additional restriction that they need to be extracted by the appropriate FSA. This is a more strict evaluation. Recall was calculated as the number of correct relations divided by the total number of relations available in the text. Only those relations that could have been captured with the described FSA were considered.

There were 267 relations (excludes the conjunctional copies) extracted from the abstracts. Overall, we achieved 62 percent recall of the described patterns and 89 percent precision. The numbers varied by FSA. The highest recall was found for the OF-FSA and the lowest for the IN-FSA where a relation was considered missing when any noun phrase introduced by “in” was missing. These relations were often extracted by another FSA but considered incorrect here. Many of the errors were due to incomplete noun phrases, e.g., a missing adjective.

To evaluate conjunctions, we counted all relations where a conjunction was part of the FSA. Conjunctions where the elements needed recombination, e.g., “breast and ovarian cancer,” were not counted since we explicitly avoid them. A conjunction was considered correct if each constituent is correctly placed in the FSA. The conjunctions were either correctly extracted (100 percent precision) or ignored. This adds a few

selective relations without introducing any errors. To learn the coverage of the FSA, we counted all occurrences of "by," "of," and "in," with a few exceptions such as "in addition," which are explicitly disregarded by the parser because they result in irrelevant relations. Seventy-seven percent of all "of" prepositions, 29 percent of all "by" prepositions, and 14 percent of all "in" prepositions were correctly captured. This indicates that the OF-FSA is relatively complete for biomedical text. The BY-FSA and IN-FSA cover a smaller portion of the available structures.

3.2.5 Ontology and Concept Space Integration

1. Additional Genescene Components

We parsed more than 100,000 PUBMED abstracts related to p53, ap1, and yeast. The parser processes 15 abstracts per second on a regular desktop computer. We stored all relations and combined them with Concept Space (Chen and Lynch 1992), a co-occurrence based semantic network, in Genescene. Both techniques extract complementary biomedical relations: the parser extracts precise, semantically rich relations and Concept Space extracts co-occurrence relations. The Gene Ontology (Ashburner et al., 2000), the Human Genome Nomenclature (Wain et al., 2002), and the UMLS were used to tag terms. More than half of the terms received a tag. The UMLS provided most tags (57 percent), and GO (1 percent) and HUGO (0.5 percent) fewer.

2. Results of Ontology Integration

In an additional user study, two researchers evaluated terms and relations from abstracts of interest to them. The results showed very high precision of the terms (93 percent) and parser relations (95 percent). Concept Space relations with terms found in the ontologies were more precise (78 percent) than without (60 percent). Terms with more specific tags, e.g., from GO versus the UMLS, were evaluated as more relevant. Parser relations were more relevant than Concept Space relations. Details of this system and study can be found in (Leroy and Chen, in press).

3.2.6 Conclusion

This study described an efficient parser based on closed-class English words to efficiently capture relations between noun phrases in biomedical text. Relations are specified with syntactic constraints and described in FSA but may contain any verb, noun, or noun phrase. On average, the extracted relations are more than 90 percent correct. The parser is very efficient and larger collections have been parsed and combined with the UMLS, GO, HUGO and a semantic network called Concept Space. This facilitates integration.

3.3 GeneScene Visualizer

The GeneScene Visualizer is a visualization tool designed to support the searching and browsing of the pathway relations extracted from PubMed abstracts by relation parsers. These relations when viewed in large quantities resemble a comprehensive knowledge map. When less common relations are sought, the interface displays genetic networks as reported in the literature. Several screenshots of the visualizer are shown in Figure 18-4. We had previously met with gene researchers and observed the manner in which they constructed pathway maps from PubMed abstracts. These pathway maps inspired the functionality of the GeneScene Visualizer. The visualizer presents the extracted relations as a gene and gene-product interaction network or map.

Architecturally, the GeneScene Visualizer is composed of two main modules: a) a relation repository and b) a visualization interface. These two modules are loosely coupled via custom-designed XML messages on top of HTTP protocol. The GeneScene Visualizer could thus interface any backend repository that presented appropriately formatted XML. The visualizer has already been used with different pathway extraction tools.

The relation repository is implemented in Microsoft SQL Server, relying on over 40 tables and thousands of lines of stored procedure code. The relation repository serves three main purposes: 1) to provide storage for over 500,000 relations extracted from PubMed abstracts 2) to provide retrieval, searching and sorting functionality for the stored relations based on user-input keywords and various ranking strategies and 3) to provide storage for the PubMed abstracts themselves, which are loaded in the visual interface when requested by the user.

The visualization interface also has three primary functions: 1) to provide a search interface for the users to retrieve relations of interest using boolean operators and partial and exact matching 2) to automatically layout the retrieved relations in a meaningful and intuitive manner via a spring embedded algorithm and 3) to show the original PubMed abstract content as reference, when the user clicks on links of interest.

The visualizer is written in Java and is launched from a web browser using Java WebStart™. The application is thus cross-platform and can be run on Windows, Apple, and Linux client machines. The application communicates with the relation repository via the HTTP protocol and XML messages in order to perform searching and fetching of the PubMed abstracts.

In addition to the general storing and visualizing of relations, the GeneScene Visualizer provides a wide range of functionalities such as relation-based searching, save and restore of user sessions, along with

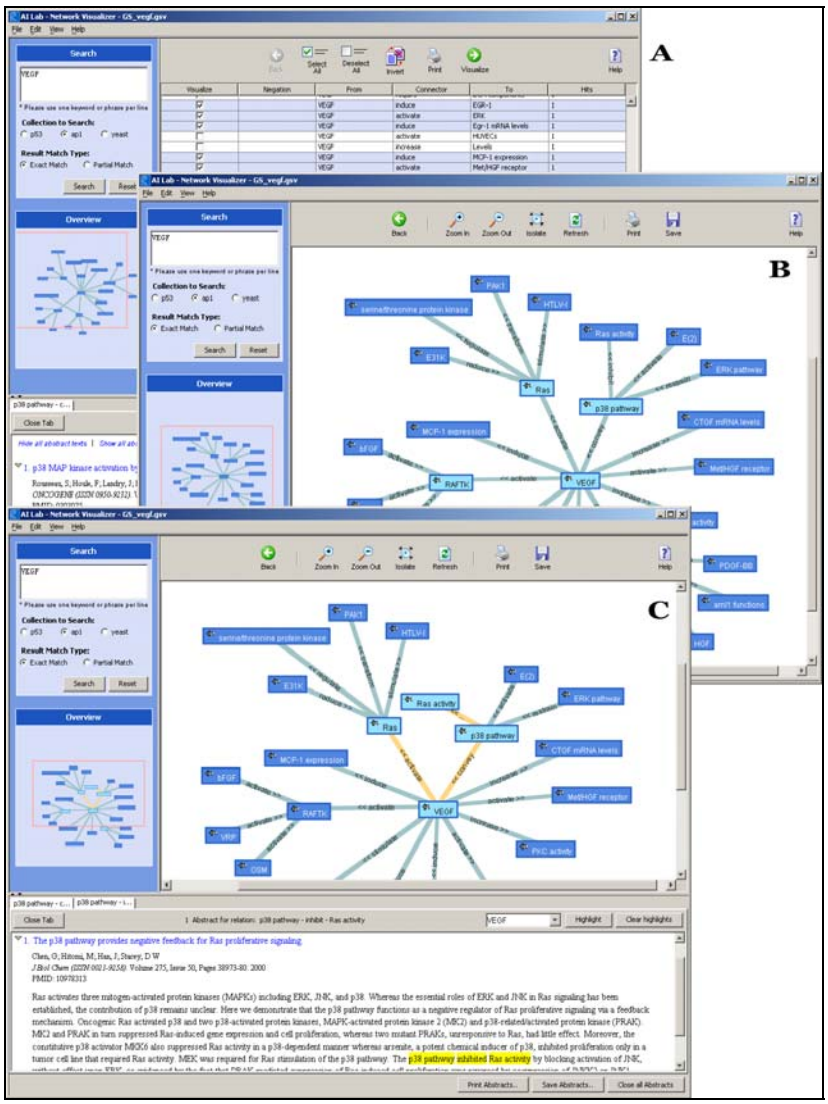

Figure 18-4. GeneScene interface. A. A user wants to search regulatory relations for VEGF (vascular endothelial growth factor), a growth factor that can stimulate tumor angiogenesis. A search in the AP1 collection by exact match results in a number of relations listed in the table in the right panel; the user then selects interesting relations (highlighted), e.g., "VEGF activate Ras", and "p38 pathway convey VEGF (signal)". B. The selected relations are visualized in a network; selected nodes ("Ras", "p38 pathway" and "RAFTK") are expanded to bring additional relations. C. Expanding the network helps the user identify more interesting relations; in this example, expanding the "p38 pathway" brings up a relation ("p38 pathway inhibit Ras activity") that seems controversial to other relations in the network ("p38 pathway convey VEGF (signal)" and "VEGF activate Ras"); by retrieving the abstracts corresponding to the relations (e.g., "The p38 pathway provides negative feedback for Ras

proliferative signaling" (Chen, Hitomi et al. 2000)), the user is able to explain the relations observed in the network (p38 provides negative feedback for Ras signaling by inhibiting Ras activity so that an equilibrium is established among VEGF, Ras and p38 pathway).

various table and network viewing manipulations of the retrieved relations. Examples include filter, sort, zoom, highlight, isolate, expand, and print. These functionalities are grouped into six categories and explained in detail below.

1. Relation searching: The interface allows researchers to search for specific elements, e.g., diseases or genes, and to view all retrieved relations as a network. Like a search engine, the GeneScene Visualizer system can take multiple keywords and can perform both "AND" and "OR" searches using keywords. A search may generate a tremendous number of matches. Thus, result ranking is necessary. We are exploring a relation ranking scheme based on various elements such as the source of the abstract, the location of the sentence in the abstract, keyword frequency, inverse abstract frequency, relation frequency, and the completeness of the relation.
2. Expanding the network: The search results returned from keyword search usually represent a very small part of the entire PubMed extracted relations. To explore related relations extracted from PubMed, the researcher can double-click on any network node and the system will retrieve relations involving that substance. By keyword searching and network expansion, researchers can find related network paths not quickly recognized through manual analysis.
3. Network presentation: To present the relation network in an easy to read format, the GeneScene Visualizer uses a spring-embedded algorithm to arrange the substances in a network. Using a physical analogy, a network is viewed as a system of bodies with forces acting between the bodies. The algorithm seeks a configuration of the bodies with locally minimized energy. With minor modification, our implementation of this algorithm layouts disconnected networks nicely. The drawback of this algorithm is that being node focused, it does not optimize the placement of links. In a complex relation network, links tend to cross and sometimes are not easy to follow.
4. Network navigation: Network navigation is similar to that found in Geographic Information Systems (GIS). The GeneScene Visualizer presents the relation network like a map. Researchers can zoom, and pan the view area of the network to gain the birds-eye view of the network and thus not miss the details. The GeneScene Visualizer also provides

researchers an overview of the network with which they can always keep a global view, providing a focus+context interface.

5. Network manipulation: The GeneScene Visualizer allows not just network navigation, but also provides the ability to manipulate the relation networks in both content and presentation. The researchers can move the elements in the network to change network layout. In addition to the network interface, relations are displayed in rows as part of a table view. From the table view, relations can be sorted, selected, and deleted. Only selected relations are included in the gene/gene products network visualization. An iterative approach to network refinement is supported.
6. Displaying relation source: By clicking on a link in the network, the GeneScene Visualizer loads the original PubMed abstract from which that particular relation was extracted. The source sentence producing the relation is highlighted to speed identification. The abstract metadata is also displayed for reference.

A demo system has been created using abstracts from the PubMed database, including relations from three test collections, which include p53 (23,243 abstracts), yeast (56,246 abstracts), and AP1 (30,820 abstracts). The demo system can be accessed at: http://genescene.arizona.edu:8080/NetVis/index.html. Preliminary user studies by cancer researchers in the Arizona Cancer Center have shown the system to effectively provide access to biomedical literature and research findings. We are currently evaluating the user interface and the visual network exploration component for their utility in exploring genetic pathway-related biomedical relations.

4. CONCLUSIONS AND DISCUSSION

We have presented two text mining systems for extracting gene-pathway relations from PubMed abstracts. The Arizona Relation Parser uses a hybrid grammar and parses sentences into their grammatical sentence structure to extract relations. The Genescene Parser recognizes noun phrases and then uses the semantics of key prepositions to anchor templates that recognize pathway relations. Both systems have achieved performance of or near 90 percent precision. Such a precision score is among the higher performing published systems for gene-pathway extraction. With a large number of precise pathway relations extracted, presenting these relations to researchers in a meaningful way becomes an important focus. We also presented a visualization tool for displaying the relations, the Genescene Visualizer. The visualizer automatically lays out selected relations in a network display for

analysis and future exploration. The network display was inspired by the manually created pathway maps researchers often make when reading PubMed abstracts and articles. We are currently evaluating the utility of the visualizer to enhance cancer researchers' access to research findings.

Having extracting and analyzed large numbers of gene-pathway relations, our future research direction has focused on consolidating or aggregating the relations. Aggregating relations combines equivalent relations, even though they may have been expressed with different words (i.e., non-mutant genes are often called wild-type genes). Such a task requires a deeper understanding of the relations. Knowing which relations say the same thing or express contradictory findings allows the visualizer to display only unique relations as well as point out possibly interesting contradictions for researchers to pursue. Such information has the potential to improve the ranking of relations returned from a search as well as to present a less cluttered gene network for visual analysis. In addition, as relations are more fully understood, they can be automatically linked to other existing knowledge sources such as the UMLS semantic network and the REFSEQ database. Expert formed relations from the database could then be merged with the automatically extracted relations to increase the accuracy and coverage of the network.

5. ACKNOWLEDGEMENTS

This research was sponsored by the following grant: NIH/NLM, 1 R33 LM07299-01, 2002-2005, "Genescene: a Toolkit for Gene Pathway Analysis."

REFERENCES

Ananko, E. A., N. L. Podkolodny, et al. (2002). "GeneNet: a Database on Structure And Functional Organisation of Gene Networks," *Nucleic Acid Research* 30(1): 398-401.

Ashburner, M., C. A. Ball, et al. (2000). "Gene Ontology: Tool For the Unification of Biology," *Nature Genetics* 25: 25-29.

Breitkreutz, B.-J., C. Stark, et al. (2003). "Osprey: A Network Visualization System," *Genome Biology* 4(3): R22.

Brill, E. (1993). A *Corpus-Based Approach To Language Learning.* Computer Science. Philadelphia, University of Pennsylvania.

Buchholz, S. N. (2002). *Memory-Based Grammatical Relation Finding.* Computer Science. Tilburg, University of Tilburg: 217.

Chen, H. And K. J. Lynch (1992). "Automatic Construction of Networks of Concepts Characterizing Document Databases," *IEEE Transactions on Systems, Man And Cybernetics* 22(5): 885-902.

Ciravegna, F. And A. Lavelli (1999). *Full Text Parsing Using Cascades of Rules: An Information Extraction Perspective.* EACL.

Daraselia, N., A. Yuryev, et al. (2004). "Extracting Human Protein Interactions from MEDLINE Using a Full-sentence Parser," *Journal of Bioinformatics* 20(5): 604-611.

Eades, P. (1984). "A Heuristic For Graph Drawing," *Congressus Numerantium* 42: 19-160.

Friedman, C., P. Kra, et al. (2001). "GENIES: a Natural-language Processing System For theExtraction of Molecular Pathways From Journal Articles," *Journal of Bioinformatics* 17(1): S74-S82.

Furnas, G. W. (1986). "Generalized Fisheye Views," in *Proceedings of the Human Factors in Computing Systems Conference* (CHI '86).

Gaizauskas, R., G. Demetriou, et al. (2003). "Protein Structures And Information Extraction From Biological Texts: thePASTA System," *Journal of Bioinformatics* 19(1): 135-143.

Hafner, C. D., K. Baclawski, et al. (1994). "Creating a Knowledge Base of Biological Research Papers," *ISMB* 2: 147-155.

Herman, I., G. Melancon, et al. (2000). "Graph Visualization And Navigation in Information Visualization: A Survey," *IEEE Transactions on Visualization And Computer Graphics* 6(1): 24-43.

Hobbs, J., D. Appelt, et al., Eds. (1996). *FASTUS: Extracting Information From Natural Language Texts. Finite State Devices For Natural Language Processing*, MIT Press.

Jurafsky, D. And J. H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, And Speech Recognition*. Upper Saddle River, Prentice Hall.

Kanehisa, M. And S. Goto (2000). "KEGG: Kyoto Encyclopedia of Genes And Genomes," *Nucleic Acid Research* 28(1): 27-30.

Kanehisa, M., S. Goto, et al. (2002). "The KEGG Databases At GenomeNet," *Nucleic Acid Research* 30(1): 42-46.

Karp, P. D., S. Paley, et al. (2002). "The Pathway Tools Software," *Bioinformatics* 18(1): S225-S232.

Kuffner, R., R. Zimmer, et al. (2000). "Pathway Analysis in Metabolic Databases Via Differential Metabolic Display," *Bioinformatics* 16(9): 825-836.

Lamping, J. And R. Rao (1986). "The Hyperbolic Browser: a Focus+Context Technique for Visualizing Large Hierarchies," *Journal of Visual Language And Computing* 7(1): 33-55.

Leroy, G. And H. Chen, in Press, "Genescene: An Ontology-enhanced Integration of Linguistic And Co-occurrence Based Relations in Biomedical Texts," *Journal of the American Society For Information Science And Technology* (Special Issue).

Leroy, G., J. D. Martinez, et al. (2003). "A Shallow Parser Based on Closed-class Words to Capture Relations in Biomedical Text," *Journal of Biomedical Informatics* 36: 145-158.

Manning, C. D. And H. Schütze (2001). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts, MIT Press.

Novichkova, S., S. Egorov, et al. (2003). "MedScan, a Natural Language Processing Engine For MEDLINE Abstracts," *Journal of Bioinformatics* 19(13): 1699-1706.

Ohta, T., Y. Tateisi, et al. (2002). "The Genia Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain," Human Language Technology Conference, San Diego, CA, USA.

Park, J. C., H. S. Kim, et al. (2001). "Bidirectional Incremental Parsing For Automatic Pathway Identification with Combinatory Categorical Grammar," Pacific Symposium on Biocomputing. 6: 396-407.

Pullum, G. K. And R. Huddleston (2002). *Prepositions And Preposition Phrases: theCambridge Grammar of the English Language*. R. Huddleston And G. K. Pullum. Cambridge, UK, Cambridge University Press.

Purchase, H. C. (2000). "Effective Information Visualization: A Study of Graph Drawing Aesthetic And Algorithms," *Interacting with Computers* 13: 147-162.

Pustejovsky, J., J. Castano, et al. (2002). "Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations," in *Pacific Symposium on Biocomputing*, Hawaii.

Rindflesch, T. C., L. Tanabe, et al. (2000). "EDGAR: Extraction of Drugs, Genes and Relations From the Biomedical Literature," in *Pacific Symposium on Biocomputing*: 517-528.

Sekimisu, T., H. Park, et al. (1998). "Identifying the Interaction Between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts," *Genome Inform*: 62-71.

Sirava, M., T. Schafer, et al. (2002). "BioMiner: Modeling, Analyzing, and Visualizing Biochemical Pathways And Networks," *Bioinformatics* 18(2): 219-230.

Thomas, J., D. Milward, et al. (2000). "Automatic Extraction of Protein Interactions from Scientific Abstracts," in *Pacific Symposium on Biocomputing*: 510-52.

Tolle, K. M. And H. Chen (2000). "Comparing Noun Phrasing Techniques for Use with Medical Digital Library Tools," *Journal of the American Society of Information Systems* 51(4): 352-370.

Tottie, G. (1991). *Negation in English Speech And Writing: A Study in Variation*, Academic Press, Inc.

Toyoda, T., Y. Mochizuki, et al. (2003). "GSCope: a Clipped Fisheye Viewer Effective For Highly Complicated Biomolecular Network Graphs," *Bioinformatics* 19(3): 437-438.

Wain, H. M., L. M, et al. (2002). "The Human Gene Nomenclature Database," *Nucleic Acids Research* 30(1): 169-171.

Yakushiji, A., Y. Tateisi, et al. (2001). "Event Extraction From Biomedical Papers Using a Full Parser," in *Pacific Symposium on Biocomputing* 6: 408-419.

SUGGESTED READINGS

Battista, G. D., et al., 1999, *Graph drawing: Algorithms for the visualization of graphs*. Prentice Hall.
This book presents a set of algorithms and approaches for displaying the structure of a graph, including planar orientations, flow and orthogonal drawing, incremental construction, layered drawings of digraphs, and spring embedder methods.

Becker, R.A., Eick, S.G., and Wilks, A.R., 1995, Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics* 1 (1), 16-28.
The focus of this article is not on displaying the structure of a network but on effectively presenting the data associated with nodes and links in the network.

Jackson, P. and I. Moulinier (2002, *Natural Language Processing for Online Applications*. Amsterdam / Philadelphia, John Benjamins Publishing Company.
This book describes the use of natural language processing for various tasks from named entity extraction to ad hoc query tasks. The book describes some of the current work from the Message Understanding Conferences (MUC).

Jurafsky, D. and J. H. Martin (2000, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, Prentice Hall.

This book presents a comprehensive treatment of natural language processing from a linguistic perspective. Many algorithms are presented for various levels of linguistic analysis.

Manning, C. D. and H. Schütze (2001, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts, MIT Press.
This book presents a statistical approach to natural language processing. The book includes both the statistics used in the models presented, but also examples where the statistical models are used.

ONLINE RESOURCES

A comprehensive list of major gene pathway analysis systems
http://ihome.cuhk.edu.hk/~b400559/arraysoft_pathway.html

An NLM funded resource for lexical information from PUBMED
http://www.medstract.org/

André Moreau and Associates Inc., English Language Resources
http://www.ajmoreau.com/english.me.html

Gene Ontology Consortium
http://www.geneontology.org/

Genia Project Home Page
http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/index.html

Human Gene Nomenclature Committee
http://www.gene.ucl.ac.uk/nomenclature/

National Library of Medicine, Entrez PubMed
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

NIH, NCI, CCR, DHHS Genomics and Bioinformatics Group
http://discover.nci.nih.gov/

The Unfied Medical Language System (UMLS)
http://www.nlm.nih.gov/research/umls/

University of Arizona Artificial Intelligence Lab, NLM funded research
http://ai.arizona.edu/go/GeneScene/index.html

QUESTIONS FOR DISCUSSION

1. What are some common approaches used to extract gene-pathway relations from text?

2. How do syntactic parsing approaches differ from semantic parsing approaches?

3. How can biomedical relational triples be used once they have been extracted from text?

4. Why do prepositions play such a key role in medical abstracts?

5. What are major issues associated with automated graph drawing?

6. What domain-specific difficulties does gene pathway visualization face?

7. What tools help meet the focus+context need of end users?