



Extracting gene pathway relations using a hybrid grammar: the Arizona Relation Parser

Daniel M. McDonald*, Hsinchun Chen, Hua Su and Byron B. Marshall

Artificial Intelligence Laboratory MIS Department, University of Arizona, 1130 E. Helen St, Tucson, AZ 85721, USA

Received on April 19, 2004; revised on July 4, 2004; accepted on July 5, 2004

Advance Access publication July 15, 2004

ABSTRACT

Motivation: Text-mining research in the biomedical domain has been motivated by the rapid growth of new research findings. Improving the accessibility of findings has potential to speed hypothesis generation.

Results: We present the Arizona Relation Parser that differs from other parsers in its use of a broad coverage syntax-semantic hybrid grammar. While syntax grammars have generally been tested over more documents, semantic grammars have outperformed them in precision and recall. We combined access to syntax and semantic information from a single grammar. The parser was trained using 40 PubMed abstracts and then tested using 100 unseen abstracts, half for precision and half for recall. Expert evaluation showed that the parser extracted biologically relevant relations with 89% precision. Recall of expert identified relations with semantic filtering was 35 and 61% before semantic filtering. Such results approach the higher-performing semantic parsers. However, the AZ parser was tested over a greater variety of writing styles and semantic content.

Availability: Relations extracted from over 600 000 PubMed abstracts are available for retrieval and visualization at <http://econport.arizona.edu:8080/NetVis/index.html>

Contact: dmm@eller.arizona.edu

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 2000 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, findings are more difficult to automatically extract than findings that appear in databases, such as Swiss-Prot, 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 hypothesis. The success of such tools, however, relies on the accuracy of the relations extracted from text. We will review existing techniques for generating biomedical relations and the methodologies used to evaluate the techniques. We then propose a relation extraction tool, the Arizona Relation Parser and present the results of an evaluation of the parser by an expert in biology. Finally, we draw conclusions and summarize our contributions.

BACKGROUND

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.

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

*To whom correspondence should be addressed.

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.* (2001) used a combinatory categorical grammar to syntactically parse complete sentence structure around occurrences of proteins. Sekimizu *et al.* (1998) used partial parsing techniques to identify simple grammatical verb relations involving seven different verbs. Yakushiji *et al.* (2001) used full syntax parsing techniques to identify not just relations between substances, but the sequence of the relations as they occurred in events. Others, while still predominantly syntactic, have incorporated different types of semantic information. Leroy *et al.* (2003) used shallow syntax parsing around three key prepositions to locate relevant relations. Thomas *et al.* (2000) reported on their system Highlight that used partial parsing techniques to recognize certain syntactic structures. 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.* (2003) that reported a 90% precision, the highest precision reported from the syntax approaches did not exceed 83%. Park *et al.* (2001) reported 80% precision. Sekimizu *et al.* (1998) reported 83% precision. Yakushiji *et al.* (2001) reported a recall of 47%. The Highlight system reported a high of 77% precision. Semantic approaches on the other hand have achieved precision as high as 91 and 96% (Friedman *et al.*, 2001; Pustejovsky *et al.*, 2002).

Despite the lower performance numbers, the evaluations of syntax approaches typically involved a larger number of documents. Park *et al.* (2001) evaluated their parser on 492 sentences, while Leroy *et al.* (2003) used 26 abstracts and Thomas *et al.* (2000) used 2565 abstracts. Using a greater number of documents in an evaluation shows the parser's performance when faced with different writing styles and topic content.

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. Rindfleisch *et al.* (2000) incorporate a greater amount of semantic information in their system EDGAR. 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% and a recall of 63%. 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.* (2002) used a semantic automaton that focused on certain verbal and nominal forms. Precision of 91% was reported along with a recall of 59%. 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.* (2002) limited their relations of interest to inhibit relations, and Hafner *et al.* (1994) and Rindfleisch *et al.* (2000) did not submit precision or recall numbers.

Balanced approaches

Balanced approaches utilize more equal amounts of syntax and semantic parsing. Syntax parsing takes place first, often resulting in an ambiguous parse. More than 100 parses can be generated for a single sentence (Novichkova *et al.*, 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 analysis are still separate. Syntax grammars remain subject to poor coverage. As semantic analysis occurs only after the 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.* (2003) reported on PASTA, a system that included complete syntax and semantic modules. The

relation extraction component of PASTA was evaluated using 30 unseen abstracts. Recall was reported at 68%, among the highest recall number published, and a precision of 65%. 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.* (2003) reported on their system MedScan which involved both syntax and semantic components. 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% coverage. In a more recent study, Daraselia *et al.* (2004) reported a precision of 91% and a recall of 21% when extracting human protein interactions from MEDLINE using MedScan. Such a high precision supports the robustness of their semantic analysis given their task. However, the recall of 21% 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 be tested over a large sample size. Adding semantic analysis to the pipe, however, did not improve the coverage of the syntax grammar.

SYSTEMS AND METHODS

In the Arizona Relation Parser, 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 purely 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 our hybrid parser 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.

We report on two main research questions in this paper. First, how can a combined syntax-semantic parsing process be implemented for biomedical texts? Second, how does our implementation of the combined syntax-semantic parser compare in precision and recall to other published systems? Our study focuses on extracting relations that contain specifically genetic regulatory pathway information. A successfully extracted relation has a gene, protein or hormone as arguments in the predicate relation as well as a relevant predicate as judged by our expert. All the information extracted is in this form of predicate (argument1, argument2). Relations extracted are used by a gene network visualizer and are intended to help researchers construct gene pathways.

Representation and rules

We use a notation adapted from Ciravegna and Lavelli (1999) to describe our parsing representation. Every lexical element a in an input sequence k is represented by a token T . Grammar rules have a binary form $(\gamma\alpha\delta, \Gamma_R)$ where $\gamma\alpha\delta$ is a non-empty string of tags and is called the rule pattern; α is called the rule core and cannot be empty; γ and δ are called the rule context and can be empty. Γ_R is a set of rule transformations that act on the rule core only. A data structure called a token chart is dynamically maintained. The token chart is a directed graph where initial vertices correspond to the lexical entities in k . Arcs represent relationships among vertices from some finite set. At first, arcs between tags represent the original order of the corresponding lexical elements from k . During the parsing process, vertices are added to the graph to reflect constituency relationships among the vertices. Figure 1 shows the token chart from the parsing process (levels 1–3) combined with a similar data structure, a knowledge pattern chart (KPC) (level 4) drawn on top. The parser outputs the top two levels of the parse chart to the relation extraction process. The relations are then extracted by using knowledge patterns on the output of the parser. The boxed numbers correspond to the output of the four major processing steps of the parser. At level 1, words are broken into their lexical tokens and tagged. Shown at levels 2 and 3 are the parses from the token chart. The output of the parsing process (level 3) is the input to the relation extraction process, which is shown in level 4. The parsing process outputs at most the top two parse layers from each sentence.

Hybrid grammar

We combined 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 1. The set of tags was chosen using three

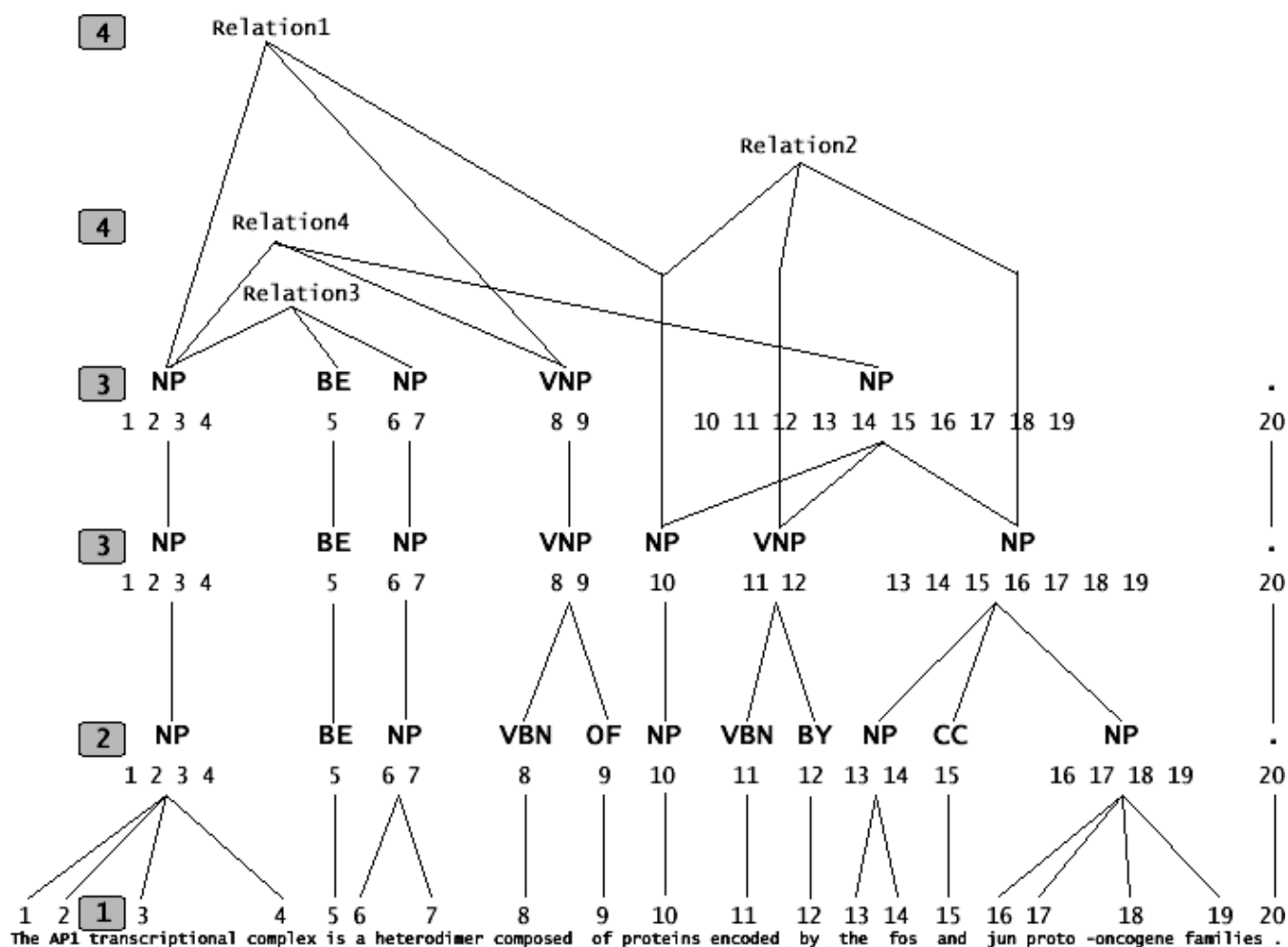


Fig. 1. A token chart (levels 1–3) with the knowledge pattern chart (level 4) applied on top.

Table 1. Sample tags from hybrid grammar

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, PRPS, RB, RBR

primary methods. First, we started with a complete lexicon extracted from the PENN TREE BANK and BROWN corpora. 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. 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. Second, domain relevant nouns were subclassed 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. Using over 150 new tags with a regular grammar eliminates the problem of over generating parses. Two different parsing rules

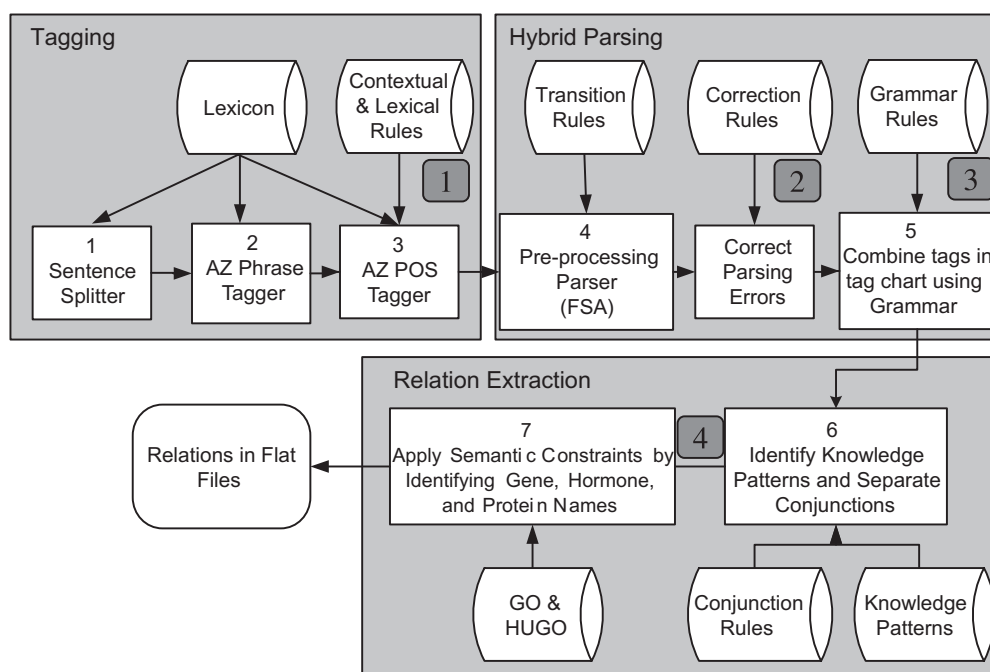


Fig. 2. Architecture diagram for the Arizona Relation Parser, consisting of three main stages: tagging, parsing and relation extraction.

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 entire parsing process now follow.

Arizona Relation Parser

The general architecture of the Arizona Relation Parser is shown in Figure 2. The boxed numbers correspond to the boxed numbers shown in Figure 1 and indicate when new arcs are added to the parse chart. Each main component found in Figure 2 will now be explained in more detail.

Sentence splitter

The parsing process begins with tokenization, where word boundaries 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).

AZ phrase tagger

The phrase tagger is based on a finite state automaton (FSA). The 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. We currently tag over 25 000 phrases.

AZ 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 over 150 new tags (generated by methods described earlier) along with the original tags from the PENN TREE BANK tag set. A mapping between the new lexical/semantic tags and the original part-of-speech tag exits so that the Brill transformation rules function as normal. The tag lexicon consists of over 150 000 entries.

Pre-process parsing

The parser performs a phrase chunking step that makes noun and verb groups using limited context. The more ambiguous parsing problems, including the handling of conjunctions and prepositions, are handled in the hybrid-parsing step. The pre-parser output appears in boxed-number 2 shown in Figure 1.

Hybrid parser

We attempt to address the poor grammar coverage problem by relaxing some of the parsing assumptions made by full parsers. Full sentence parsing up to a root node of a binary branching tree is not necessary because we are interested in extracting knowledge patterns (semantic triples). The hybrid parser 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. The input to the hybrid parser is

```

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

```

Fig. 3. A parsing rule with a rule pattern and transformation.

the pre-parsing output. Internally, a cascade of four 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 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 pre-processing requirement.
- Level 4: Relative and subordinate clauses are recognized.

The output from both levels 3 and 4 are then passed to the relation extraction step. An example of the parsing output is shown in Figure 1, labeled with the boxed number 3.

The hybrid parser utilizes a regular grammar that handles dependences 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 3 gives an example of a grammar rule. The rule pattern in Figure 3 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 follow a 'BY' tag and be followed by a '.' tag to be combined into a new noun phrase. The GRAMMAR LEVEL designation, in Figure 3, refers to which cascade of the four uses this rule. There are a total of 1778 grammar rules applied in the four cascades. Currently 843 of those rules are unique, while the others apply in two or more levels. Figure 4 shows several of those rules with the rule core in bold face and the rule context 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

```

Fig. 4. Parsing rules with rule core bolded.

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. The relations extracted are stored in a directed graph called a knowledge pattern chart (KPC). Similar to the parsing rules, knowledge patterns consist of rule patterns ($\gamma\alpha\delta$), with rule context (γ, δ) and rule core (α) and transformations (Γ_R) that are applied 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. Samples of the roles are defined below.

- Role 0: The tag plays no role.
- Role 1: The tag fills argument slot 1.
- Role 2: The tag acts as the predicate.
- Role 3: The tag fills argument slot 2.
- Role 6: Captures nominalization, separated by an 'of' (Example 1) or a gerund verb (Example 2) and occur in prepositions, unlike Role 8.

Example 1. [for involvement of c-Abl/FOR] [in recombinational repair of DNA strand breaks/INN].

Example 2. [K12/NP] [may increase/VP] [aggressiveness/NP] [not by altering proliferative pathways/HOW].

- Role 8: Captures verbs in nominal form (Example 1) and also agentive form (Example 2) when they appear in noun phrases.

Example 1. [induction of c-myc expression/NP] [by/BY] [GM-CSF/NP].

Example 2. [The Mdm2 oncoprotein/NP] [is/BE] [a potent inhibitor of p53/NP].

```

<KNOWLEDGEPATTERN NUM="1">
<RULEPATTERN>
<PREVIOUSCONTEXT />
<RULECORE>NP BE NP VNP NP VNP NP</RULECORE>
<FUTURECONTEXT />
</RULEPATTERN>
<TRANSFORMATION>
  <RELATION SEQ="1">
    <SLOT ROLE="1" TOKEN="1" />
    <SLOT ROLE="2" TOKEN="2" />
    <SLOT ROLE="3" TOKEN="3" />
  </RELATION>
  <RELATION SEQ="2">
    <SLOT ROLE="1" TOKEN="1" />
    <SLOT ROLE="2" TOKEN="4" />
    <SLOT ROLE="3" TOKEN="5" />
  </RELATION>
  <RELATION SEQ="3">
    <SLOT ROLE="1" TOKEN="5" />
    <SLOT ROLE="2" TOKEN="6" />
    <SLOT ROLE="3" TOKEN="7" />
  </RELATION>
</TRANSFORMATION>
</KNOWLEDGEPATTERN>

```

Fig. 5. A knowledge pattern rule.

Sentences may contain multiple overlapping knowledge patterns, with tags playing multiple roles. Figure 5 shows the knowledge pattern rule that extracted the relations in Figure 1. The parsing output, ‘NP BE NP VNP NP VNP NP’, matched the rule pattern from Figure 5. Given a match of the rule pattern, the rule assigned roles to the rule core tags.

Knowledge pattern parsing increases grammar coverage because it ignores the majority of prepositional attachment. Some knowledge patterns (such as patterns involving roles 6 and 8) probe into prepositional phrases, but the majority do not. A one-level clause embedding limit also simplifies parsing (example ‘p53-induced cell death involves a Bax-dependent caspase-3 activation’). In addition, parse trees need not combine into a single root node so main clauses are not distinguished from subordinate clauses.

Conjunctions

When Role 1 or Role 3 contains a conjunction of noun phrases, the noun phrases are split after the relation identification phase. For example, the relation induce (p53, both growth arrest and apoptosis) would be split into the following two relations: induce (p53, growth arrest) and induce (p53, apoptosis).

Applying semantic constraints

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, neutralize, phosphorylat, repress, trigger and induc. Our biology expert generated the list of relevant verb stems by examining verbs appearing in PubMed.

RESULTS

The goal of the relation parser design was to increase the recall and precision of relation extraction by combining semantic and syntax analysis. The performance of the hybrid grammar together with the semantic filtering was tested in an experiment involving 100 unseen abstracts. A total of 50 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, a PhD 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)’. The hybrid parser 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.

Relations of category A and B had been previously extracted in other research and thus our primary evaluation is based on this grouping of relations. We are analyzing the utility of category C relations in ongoing research. The primary results from the experiment are listed in Table 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%. When we widened the pool of correct relations

Table 2. Parser performance results

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

Table 3. Parser performance less filtering errors

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

Table 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

to include pathway-relevant relations (category C), correct relations jumped to 116 producing an 89% precision number, as shown in Table 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.

To perform the recall experiment, the 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. Of the abstracts, 14 produced no relations. Recall equaled the ratio of system-identified relations to the expert identified relations. Table 2 shows the system's recall score of 35%. In addition to the standard recall score, we wanted to show the recall of the parsing component unaffected by semantic filtering. Table 3 shows the parser extracted 61% of the relations. Thus, 26% of the correctly extracted relations were removed by the filtering process.

Including the errors introduced by the semantic filtering component, Table 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% 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% of the expert identified relations due to parsing errors.

DISCUSSION

When including biologically relevant relations (category C) in the total of correct relations, the precision of the parser (89%) is among the top performers that we reviewed. Such performance provides substantial support for the effectiveness of the hybrid grammar. However, when including just categories A and B as correct relations, as is common in other evaluations, the parser's precision dropped significantly. We attribute this drop to the lacking sophistication of our semantic filtering approach. Our semantic filtering approach represents an efficient way to utilize current resources in the bioinformatics community, such as GO and HUGO, to approximate the identification of gene and gene products within noun strings. Such an approach will be replaced by more accurate algorithms for matching biological entities.

The recall of gene-pathway relations before semantic filtering was 61%. This total also ranks among the top reported recall numbers for biomedical relation extraction. To our knowledge only Friedman *et al.* (2001) and Gaizauskas *et al.* (2003) have reported a higher recall number. The high performance of the pre-filtered recall number attests to the coverage of our hybrid grammar. Such coverage was shown over a large number of abstracts (50) and without any verb restrictions, such as extracting only inhibit relations. Despite the coverage of the hybrid grammar, however, a number of good relations were removed at the semantic filtering stage. The recall total after semantic filtering was 35%. The large decrease in recall performance relates again to the poor performance of the semantic filtering step. The 'semantic constraints' described earlier that we applied turned out to be too strict in that many relevant relations identified were filtered out before extraction. The next large decrease in recall was a result of missing extraction rules that could have theoretically extracted the correct relation. While the coverage of the parser represents a strong point compared to recall numbers of other published systems, we expect more extraction rules to improve performance. A final positive outcome of the experiment was the low number of parsing errors, which totaled 2.7%. The low ambiguity of the grammar is a result of the increased number of semantic tags along with the grammar being specified to include rule context along with rule core.

CONCLUSIONS

The use of a hybrid grammar for biomedical relation extraction has shown promise for extracting relations with high recall and precision. The hybrid grammar seeks to decrease the ambiguity of syntax grammars, and at the same time increase the coverage of the grammar above a purely semantic approach. With parsing errors reported at 2.7%, the grammar has been shown to be quite accurate. The hybrid grammar also performed well-extracting biologically relevant relations with a precision of 89% and a recall before semantic filtering of 61%. The semantic filtering of the relations turned out to be less precise than expected. Recall of the relations declined to 35% after applying semantic filtering. As a result, more sophisticated algorithms are required to identify biological entities of interest in order to improve the recall and precision of the relation parser. Future directions for this research include such an undertaking.

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

- Brill, E. (1993) *A Corpus-Based Approach to Language Learning*, Computer Science. University of Pennsylvania, Philadelphia.
- Buchholz, S.N. (2002) *Memory-Based Grammatical Relation Finding*, Computer Science. University of Tilburg, Tilburg. pp. 217.
- Ciravegna, F. and Lavelli, A. (1999) Full text parsing using cascades of rules: an information extraction perspective. *Proceedings of the ninth conference on European chapter of the Association for computational Linguistics*. Morgan Kaufmann, Bergen, Norway, June 8–12.
- Craven, M. (1999) Learning to extract relations from MEDLINE. *Proceedings of AAAI Workshop on Machine Learning for Information Extraction*. AAAI Press, Orlando, FL.
- Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A. and Mazo, I. (2004) Extracting human–protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics*, **20**, 604–611.
- Friedman, C., Kra, P., Yu, H., Krauthammer, M. and Rzhetsky, A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **17**, S74–S82.
- Gaizauskas, R., Demetriou, G., Artymiuk, P. and Willett, P. (2003) Protein structures and information extraction from biological texts: the PASTA system. *Bioinformatics*, **19**, 135–143.
- Hafner, C.D., Baclawski, K., Futrelle, R.P., Fridman, N. and Sampath, S. (1994) Creating a knowledge base of biological research papers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **2**, 147–155.
- Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M. and Tyson, M. (1996) FASTUS: Extracting information from natural language texts. In Roche, E. and Schabes, Y. (eds), *Finite State Devices for Natural Language Processing*. MIT Press Cambridge, MA.
- Jurafsky, D. and Martin, J.H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River.
- Leroy, G., Chen, H. and Martinez, J.D. (2003) A shallow parser based on closed-class words to capture relations in biomedical text. *J. Biomed. Inform.*, **36**, 145–158.
- Novichkova, S., Egorov, S. and Daraselia, N. (2003) MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, **19**, 1699–1706.
- Ohta, T., Tateisi, Y., Hideki, M. and Jun'ichi, T. (2002) The genia corpus: an annotated research abstract corpus in molecular biology domain. *Proceeding of Human Language Technology Conference*. San Diego, CA, USA, 489–493.
- Park, J.C., Kim, H.S. and Kim, J.J. (2001) Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. *Pac. Symp. Biocomput.*, **6**, 396–407.
- Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M. and Cochran, B. (2002) Robust relational parsing over biomedical literature: extracting inhibit relations. *Pac. Symp. Biocomput.*, 362–373.
- Rindfleisch, T.C., Tanaber, L., Weinstein, J.N. and Hunter, L. (2000) EDGAR: extraction of drugs, genes and relations from the biomedical literature. *Pac. Symp. Biocomput.*, **5**, 517–528.
- Sekimizu, T., Park, H. and Tsujii, J. (1998) Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. *Genome Inform.*, **9**, 62–71.
- Thomas, J., Milward, D., Ouzounis, C., Pulman, S. and Carroll, M. (2000) Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput.*, 541–552.
- Yakushiji, A., Tateisi, Y., Miyao, Y. and Tsujii, J. (2001) Event extraction from biomedical papers using a full parser. *Pacif. Symp. Biocomput.* **6**, 408–419.