Pacific Symposium on Biocomputing 6:396-407 (2001)

BIDIRECTIONAL INCREMENTAL PARSING FOR AUTOMATIC PATHWAY IDENTIFICATION WITH COMBINATORY CATEGORIAL GRAMMAR[a]

Jong C. Park*, Hyun Sook Kim, Jung Jae Kim
Computer Science Department and
Advanced Information Technology Research Center
Korea Advanced Institute of Science and Technology (KAIST)
373-1 Kusong-dong, Yusong-gu, Daejon 305-701 South KOREA
{park,hskim,jjkim}@nlp.kaist.ac.kr

As the importance of automatically extracting and analyzing various natural language assertions about protein-protein interactions in biomedical publications is recognized, many uses of natural language processing techniques are proposed in the literature. However, most proposals to date make rather simplifying assumptions about the syntactic aspects of natural language due to various reasons including efficiency. In this paper, we describe an implemented system that utilizes combinatory categorial grammar known to be competent in modeling natural language, with a controlled mechanism for the parser to operate bidirectionally and incrementally. We discuss the performance of the system on a large set of abstracts in MEDLINE with quite encouraging results.

1 Introduction

The need for automating the process of extracting and analyzing natural language assertions about protein-protein interactions in biomedical publications is well recognized, given the ever increasing volume and importance of the archived collection of natural language documents in databases such as MEDLINE. The field has two complementary aspects from the natural language processing perspective. On the one hand, the terminology is not only immense already but also rapidly growing, challenging the use of a pre-defined lexicon alone. On the other, the type of interactions is relatively fixed, where one can find corresponding natural language predicates in a pre-determined manner.

Proposals in the literature often make simplifying assumptions about either the terminology or the type of interactions. For example, Blaschke et. al.[1] simply assume that protein names are specified by the user, and focus instead on extracting the type of interactions among such proteins. In Rindflesch et. al.[2], much attention has been paid to the heuristic recovery of drug, gene and cell names from the natural language context, leaving the specific type of interactions unaddressed. Likewise, Stapley and Benoit[3] show an approach to

[a] The work was funded by the Korea Science and Engineering Foundation through AITrc.

identifying meaningful gene terms, before constructing a graph of those terms so that they are linked to one another if they "occur together in the same document with statistically significant frequency," leaving out further details of such relations as described in the relevant documents.

Interestingly, none of the above approaches look seriously into the syntactic structure of the given natural language sentence. For instance, in extracting the desired relation `proteinA-action-proteinB`, Blaschke et. al.'s approach finds it sufficient to locate the pattern `X1 proteinA X2 action X3 proteinB X4`, where any string may come as `X1` through `X4`, as in their example 3a2: "spatzle acts immediately upstream of the membrane protein toll in the genetic pathway, suggesting that **spatzle** could encode the ventrally localized ligand that **activates** the receptor activity of **toll**." While this approach may sometimes incorrectly extract relations that are not justified by the document under consideration, as in the example shown above, and other times miss relations that are present, the justification is that its performance is still impressive because it operates under the assumption that if the fact is common and thus useful, it will be present in the same collection of abstracts in a more palatable form.

Researchers tend to favor simple patterns, instead of more complex grammars, for the description of natural languages that are believed to be in the class of (mildly) context-sensitive languages in Chomsky hierarchy. The belief is that, while a more sophisticated grammar formalism would give rise to a much higher precision in extracting the desired relations, the parsing complexity will consequently rise and the daunting size of the relevant database makes it simply impossible to overlook the performance issues. This problem is well understood in natural language processing communities; the reader is referred to Hobbs et. al.[4] for illuminating discussion on this matter, especially with respect to the U.S. DARPA Message Understanding Conferences (MUCs), as to why it is considered reasonable to use regular grammars to model natural language. For one, experiences with the MUCs assured researchers in the field that the task of information extraction does not require full text understading capabilities, which would, one might guess, include the incorporation of a fully adequate grammar formalism for natural language.

Both Thomas et. al.[5] and Humphreys et. al.[6] describe approaches to information extraction from biomedical publications utilizing the techniques for MUCs, using grammar formalisms low in Chomsky hierarchy. In particular, Thomas et. al. describe a general-purpose IE engine, Highlight, that incorporates techniques used by FASTUS (cf. Hobbs et. al.[4]), most notably cascaded finite state machines to group sequences of words into phrases of various types, along with a method of ranking templates. They analyzed around 200 abstracts by hand to tune the engine to the biomedical domain, and tested the system by analyzing 2565 unseen abstracts, reporting the overall performance of precision in the range of 69 to 77, depending on the measures.

In this paper, we describe a system that parses abstracts in MEDLINE and extracts information about protein-protein interactions, using Perl and Prolog to implement the CKY-based parser for a combinatory categorial grammar (CCG) that is known to fully characterize the syntactic (and other) aspects of natural language, in this case English (cf. Steedman[7]). We have tuned our general-purpose lexicon to the set of 64 "difficult" sentences initially supplied to us, and subsequently tested the system on around 250,000 unseen abstracts on *cytokine*, published between winter 1963 and May 2000, in MEDLINE. We find that the precision is currently in the range of 80, with much further room to improve because of the adequate expressive power of CCG for modeling natural language. It takes 3 minutes on a Sparcs Enterprise 250 to process the initial set of sentences, and about 13 minutes to select and process 200 sentences, in the unseen set of data. The key idea to gaining the favorable performance is that the system first jumps into the verbs corresponding to any of the pre-compiled list of interaction types, and somewhat liberally scans their neighbors from there on bidirectionally using a regular grammar to propose NP candidates, with the CCG parser incrementally validating them.

The rest of the paper is organized as follows. Some interesting aspects of the natural language data are shown in Section 2. We introduce a version of combinatory categorial grammar for English in Section 3. Our system is described in Section 4. The results are discussed in Section 5.

2 Data Analysis

In this section, we restrict our attention to verbs of the kind shown in (1) below, along with their inflected forms and related noun forms.

(1) a *activate*, *accelerate*, *augment*, *induce*, *stimulate*, *require*, *up-regulate*
b *inhibit*, *abolish*, *block*, *down-regulate*, *prevent*

We use the notations '$\mathtt{X}\smile\mathtt{Y}$' and '$\mathtt{X}\frown\mathtt{Y}$' for the relations '`X pos Y`' and '`X neg Y`', respectively, where `pos` and `neg` are predicates shown in (1) a and b, respectively. The notation '`...`' abbreviates what is obvious. We note that our proposal does not take advantage of the specific nature of the interactions.

2.1 Coordination

(2) shows verb phrase (VP) coordination. To explain the obvious, both *inorganic phosphate* and *HPr kinase* precede the verb *activated*, but the syntax

dictates that it is only the former that works as the subject.

(2) Inorganic phosphate *inhibited HPr kinase* but *activated HPR phosphatase.*

The relations to be extracted from (2) are then `inorganic phosphate`⌢`HPr kinase` and `inorganic phosphate`⌣`HPR phosphatase`.

In (3), the underlined coordination item may be related to many pairs of conjuncts, but the use of hyphens ('`-`') gives a clue to the intended pairing. The relation to be extracted is: `forskolin`⌣`the phosphorylation of ... macrophages`. Notice that the object NP so identified is still internally ambiguous.

(3) Interestingly, under the same condition, forskolin (20 mumol/L) stimulated the phosphorylation of *LPS- and PMA-triggered* p38 MAPK of murine peritoneal suppressor macrophages, suggesting that activation of p38 MAPK is regulated positively by *both PKC and PKA.*

In contrast to (2) and (3), (4) contains conflicting clues to coordination.

(4) All vasodilators activated *K-Cl cotransport in LK SRBCs and HYZ in VSMCs*, and this activation was inhibited by calyculin and genistein, two inhibitors of K-Cl cotransport.

The underlined coordination item *and* may be syntactically interpreted to relate any of the following pairs, even if we assume that *K-Cl* and *cotransport*, and *LK* and *SRBCs* as well, are inseparable.

(5) a [K-Cl cotransport in LK SRBCs] and [HYZ in VSMCs]
b [LK SRBCs] and [HYZ]
c [LK SRBCs] and [HYZ in VSMCs]

If the preference is given to structural similarity and/or complex structure for coordination, (5) a would be the first guess, but the context, supported by the repetition of *K-Cl cotransport* as underlined, disfavors (5) a, since *K-Cl cotransport* must not be buried in the coordinate structure for this interpretation. It is clear that decisions of this kind are very hard to make one way or the other, especially when one considers only the syntactic information.

2.2 Appositions and Compound Nouns

The extraction of the relation `mepacrine`⌢`PLA2` from (6) is well justified, as it is what is presupposed by the utterance, though not explicitly asserted.

(6) *Mepacrine, a PLA2 inhibitor*, prevented HO-1 induction by cytokine.

(7) shows a slight variation to the appositive structure in (6).

(7) jNK2AS treatment induced the expression of *the CDK inhibitor p21* in parental MCF-7, RKO, and HCT116 cells.

In contrast, (8) does not justify the extraction of the relations `toxins`⌢`phosphatase` and `caspase-3`⌣`apoptosis`, as the structure of the compound nouns supports neither of the claims that all toxins inhibit phosphatase and that all apoptosis are dependent upon caspase-3.

(8) *Phosphatase-inhibiting toxins* can induce *caspase-3 dependent apoptosis* in an untrarapid manner by altering protein phosphorylation.

Incidentally, the extraction of the relation `TGF-1`$\smile$`ERK` from (9) is apparently justifiable. Nevertheless, this prediction is not as reliable as the corresponding ones for (6) and (7), since it depends on the mode and nature of the main verb, in this case *suggests*. For instance, one can think of the sentence "*The matching response of ERK activation to TGF-1 in SHR cells could not be ascertained,*" which does not endorse such a relation.

(9) The matching response of ERK activation to TGF-1 in SHR cells suggests that the MAP_KINASE-signaling pathway remains largely unchanged in the regulation of vascular smooth muscle growth by TGF-1 in spontaneously hypertensive rats.

2.3 Anaphoric Expressions

Pronominals such as *they*, *it*, and *all* are used to refer to expressions explicitly mentioned earlier (or implied by the preceding discussion), as in (10).

(10) Zinc, OKADAIC_ACID, CALYCULIN_A, cantharidin, and the caspase inhibitor z-VAD-fmk, *all* prevented the cleavage of D4-GDI, DNA digestion, and apoptosis.

Also, demonstratives such as *this* and *that* are utilized to refer to certain expressions, with the added notion of closeness (or distance) for the clarity of reference, as in (11).

(11) We demonstrate that proliferation of embryonic multilineage hematopoietic progenitors is also regulated by a hypoxia-mediated signaling pathway. *This pathway* requires HIF-1 (HIF-1alpha/ARNT heterodimers) because Arnt (-/-) embryoid bodies fail to exhibit hypoxia-mediated progenitor proliferation.

Since the relation `HIF-1`$\smile$`this pathway` is not of much use in this form, it is clear that such referring expressions must be resolved properly. Sometimes it is relatively simple to resolve such referring expressions, as in (11), where there is a uniquely matching expression in the previous sentence. (4) shows another example of the kind.

As expected, such a resolution algorithm must be quite resourceful in general, however, e.g. in order to determine what is considered protected in (12).

(12) In separate hearts, anisomycin mimicked the anti-infarct effect of PC, and *that protection* was abolished by genistein.

2.4 Summary

The foregoing analysis is necessarily brief. We summarize our findings.

(a) Coordination: We find many uses of coordination in the abstracts, perhaps due to the fact that it provides the needed conciseness of expression, as well as the additional information of contrast. This is one of the constructions where we can get much useful information if correctly handled, but it requires non-trivial heuristics to identify the intended coordination at all times (cf. Park and Cho[8]).

(b) Appositions and Compound Nouns: Appositions are also used quite often. The relevant information can be extracted in a relatively cost-effective and safe way, although the information thus extracted may not usually be novel and useful. Compound nouns may sometimes carry some presuppositional information that can be extracted as a standalone fact. However, moderate attention should be paid to the particular syntactic structure, since this is not always guaranteed.

(c) Anaphoric Expressions: Anaphoric expressions usually work to thread multiple sentences. They are often used to balance the information weight across sentences, and much useful information is gained if they are properly resolved. However, this is where a sentence grammar such as combinatory categorial grammar is not of much help.

3 Combinatory Categorial Grammar

Combinatory categorial grammars (CCGs) are combinatory extensions to pure applicative categorial grammars (CGs) that are originally conceived by Ajdukiewics in 1935 and further developed by Bar-Hillel in 1953. CCGs are actively studied grammar formalisms in linguistics, computational linguistics, and natural language processing. They are rapidly gaining recognition among the researchers in the field, due primarily to the fact that the languages that they model are proved to belong properly in the class of (mildly) context-sensitive languages, which includes natural languages such as English, along with their cousin grammar formalisms such as linear indexed grammars and tree adjoining grammars, in addition to the fact that they operate in a surprisingly intuitive way. It is best to explain CCG with examples.

(13)

$$\begin{array}{cccl}
\text{Inorganic phosphate} & \text{inhibited} & \text{HPr kinase} & \\
\cline{1-1}\cline{2-2}\cline{3-3}
np & (s\backslash np)/np & np & \\
\cline{2-3}
 & \multicolumn{2}{c}{s\backslash np} & > \\
\cline{1-3}
\multicolumn{3}{c}{s} & <
\end{array}$$

Transitive verbs like *inhibited* are assigned the category $(s\backslash np)/np$, which "expects" a phrase of category np on its right (the second occurrence of np; the directionality is indicated by the slash symbol $/$, leaning to the right) and then "expects" another phrase of category np on its left (the first occurrence of np; notice that the backslash symbol $\backslash$ is leaning to the left), to give rise to the phrase of category s, which corresponds to a (grammatical) sentence in English. Such a computation is done by function application as indicated by the symbols $>$ (forward function application) and $<$ (backward function application) in the derivation, in the sense that the category X/Y is a function that expects the category Y as its argument, and when applied to the category Y, it gives rise to the result category X. (14) shows another example, where *co* indicates that the corresponding word is a coordinating item and the symbol $<\Phi^n>$ indicates the particular derivation involving the coordinate structure.

(14)
$$\begin{array}{cccccc|l}
\text{Inorganic phosphate} & \text{inhibited} & \text{HPr kinase} & \text{but} & \text{activated} & \text{HPR phosphatase} & \\
\cline{1-1}\cline{2-2}\cline{3-3}\cline{4-4}\cline{5-5}\cline{6-6}
np & (s\backslash np)/np & np & co & (s\backslash np)/np & np & \\
\cline{2-3}\cline{5-6}
 & \multicolumn{2}{c}{s\backslash np} & & \multicolumn{2}{c}{s\backslash np} & > \\
\cline{2-6}
 & \multicolumn{5}{c}{s\backslash np} & <\Phi^n> \\
\cline{1-6}
\multicolumn{6}{c}{s} & <
\end{array}$$

Examples in (13) and (14) utilize only function application, and are characterizable by a CG. In addition to function application, CCGs utilize a limited set of combinators, such as **T** (type raising), **B** (function composition), and **S** (function substitution), to allow certain pairs of consecutive phrases to combine with each other for a larger phrase. The idea is that in order to model the syntactic aspects (or grammaticality measures) of a particular natural language, such as English, it suffices to define its lexicon, which includes the allowed categories for each lexical item (or word), and its stock of combinators, for the grammatical combination of neighbor phrases. Further details are beyond the scope of the present paper (cf. Steedman[7]).

4 Bidirectional Incremental Parsing

The key idea to the bidirectional incremental parsing, as explained in Section 1, is that the system is designed to jump into the verbs corresponding to any of the pre-compiled list of interaction types, or the *target* verbs, and to scan the neighbors of the target verbs bidirectionally and incrementally, with the CCG parser evaluating the proposed NP candidates to see if the string of words is indeed combinable as a single NP, until the parser is told to stop looking further for the particular verbs under consideration. The process repeats for all the other target verbs in the the same sentence before it moves on to the next sentence. The procedure is described in further detail below.

Table 1: Frequencies of part-of-speech tagged entries in the POS tag lexicon

POS tag	frequency	POS tag	frequency
pronoun	0.00116	adverb	0.05458
determiner	0.00054	verb	0.13460
foreign word	0.00592	coordination item	0.00023
preposition	0.00248	wh-word	0.00062
cardinal number	0.00318	adjective	0.18062
modal	0.00063	noun	0.61049

We first describe the procedure for the part-of-speech (POS) tag and CCG category assignment to lexical items.

(a) During the initialization stage, the information about the neighbors of keywords in the form of a regular grammar is parsed into the system. Special symbols such as punctuation marks are also identified and isolated.

(b) Using the keywords as anchors, those sentences that contain them are extracted from the collection of abstracts already identified from the bibliographic databases, such as MEDLINE. All the words in the matching parentheses are grouped together into an atomic sequence, effectively treated as a single word tagged as a noun (or NN) by the system from this time on.

(c) Each word in the sentence(s) is assigned a POS tag or POS tags. For the present purpose, the first POS tag for each word is used as its representative POS tag. All the words that are not sentence-initial and start capitalized are POS tagged simply as NN. All the unknown words, i.e. those not in the lexicon, are POS tagged as NN. There are about 100,000 entries in the POS tag lexicon. The frequencies of the POS tags in the lexicon are as shown in Table 1.

(d) Each word in the sentence(s) is also assigned a CCG category or CCG categories. At this step, all the CCG categories defined in the CCG lexicon are assigned to each word. If a given word is not listed in the CCG lexicon, it is treated as a noun and assigned the CCG categories `n`, `n/n`, `np/n`, and `np`. Most functional words, such as prepositions, as well as adjectives and adverbs, are defined in the CCG lexicon. There are about 36,000 lexical entries in the current CCG lexicon.

Based on the POS tags and CCG categories assigned to the lexical items, the system repeats the process of proposing NP candidates, utilizing the regular grammar, and validating them, utilizing the CCG parser. The following procedure starts with each keyword (KW) in each rule of the regular grammar and scans its leftward and rightward neighbors. A sample rule for passive structures in English is shown below, where the *by*-phrase is explicitly present.

(15) `NP_A+BeV+KW+RB*+by+NP_B`

As the match for lexical items, such as `by`, and the match for POS tags, such as `BeV` (*is*, *was*, *are*, *were*, etc), are straightforward, the details will not be described here. The pattern `RB*` in the example rule indicates that any number, including zero, of adverbs and adverbial phrases may come in its place. The following procedure is concerned with the search for NPs, as in `NP_A` and `NP_B`. As the search is not always from left to right, we need to distinguish the case when the NP is searched to the left of the keyword (←) and the case when it is searched to the right of the keyword (→). Due to space, we will not provide further justifications for the conditions. Notice also that NP candidates are proposed initially in a liberal manner.

(a) Propose NP candidates

→: If the present word is NN POS-tagged, and the next word on its right is not NN POS-tagged, collect the words including the present one and propose the sequence as an NP candidate. Otherwise keep collecting the words.

←: If the present word is DT POS-tagged, or if the present word is JJ or NN POS-tagged and the next left word is not any of the DT, JJ, or NN, collect the words including the present one and propose the sequence as an NP candidate. Otherwise keep collecting the words.

(b) Give up the search for NP candidates.

→: When '`that`', '`into`', `VBG` (present participle), `W` (wh-word), or '`in`' (except when it is preceded by *increase* or *decrease*) is found.

←: When '`, only`' is encountered; when `NN` is not found to the left of '`VB that`'; or when '`,`' is not found to the (non-immediate) left of '`, and`'.

← or →: When '`:`' is found; when the word not in the list of '`,`', '`and`', '`or`', `DT`, `JJ`, or `NN` is found or '`,`', '`and`', or '`or`' is found in a sequence (as in '`, and`', '`, or`') though the word we are looking for is `NP1` (`baseNP` or its coordination); or when '`,`' is found and the next word is either `VB` or `RB+VB`.

(c) Disregard the proposed NP candidates.

←: after '`,`' is found, the next word is IN (preposition) POS-tagged.

Other conditions that extend the search for NP candidates are omitted due to space. These NP candidates are subsequently validated by the CCG parser, which checks to see if the resulting category is `np`. As we can see from example (3), an NP with an unambiguous boundary may still be structurally ambiguous. While a further disambiguation process is not described in this paper, the CCG parser certainly identifies this ambiguity. The full range of ambiguous readings must be narrowed down for a finer-tuned generation of the argument NPs than those shown in the next section.

5 Results

5.1 Seen Data

The output of the system is shown below with a preceding `=>`. The self-evident format is designed to facilitate the process of populating a database.

(16) Forskolin, a direct activator of ADENYLYL_CYCLASE also stimulated ERK and P38 activities in these cells suggesting the involvement of cAMP in this process.

```
=> activate(Forskolin,ADENYLYL_CYCLASE)
=> activate(Forskolin,ERK&P38)
```

In (16), the first relation is due to the apposition, and the second due to the main clause verb. In (17) below, the two relations are due to the presupposition(s) associated with the compound nouns.

(17) Thus, these data suggest that activation of ERK by calcitonin gene-related peptide involves a H89-sensitive KINASE A and a wortmannin-sensitive PI3-kinase while activation of p38 MAPK by calcitonin gene-related peptide involves only the H89 sensitive pathway and is independent of PI3 kinase.

```
=> activate(calcitonin gene-related peptide,ERK)
=> activate(calcitonin gene-related peptide,p38 MAPK)
```

The third argument position shown below specifies the catalytic information.

(18) We previously showed that arginine vasopressn (AVP) stimulates heat shock protein 27 (HSP27) induction through PKC activation in aortic smooth muscle A10 cells.

```
=> activate(arginine vasopressin,heat shock protein 27,PKC activation)
```

5.2 Unseen Data

The system was tested on around 250,000 abstracts on *cytokine* in MEDLINE. In order to measure the performance of the implemented system, we have tried several methods of selecting the set of sentences. In one of the methods, the system went through the abstracts, starting from the most recent ones (May 2000), until the system was able to extract 182 relations, out of 492 sentences with relevant keywords.[b] The recall and precision rates, 48 and 80, respectively, were computed by hand. The other methods produced similar results. Table 2 shows the distribution of incorrectly extracted relations from the method explained above. Note that we did not consider that the full resolution of anaphoric expressions is a relevant task.

The 'wrong subject' category in Table 2 includes errors of the kind shown in (19). This is due to the missing CCG category $(s\backslash np)/np$ (transitive verb) for *bound*, and can be addressed by adding the corresponding entries to the

[b] Sentences without both of the two arguments were simply disregarded.

Table 2: Distribution of incorrectly extracted relations from unseen data

reason	#	reason	#
wrong coordination	13	wrong subject	7
wrong apposition	3	wrong CCG category	4
wrong POS tag	1	wrong object	4
negation	2	gerund as subject	1
A(B) as apposition	1	wrong sentence boundary	1

CCG lexicon. Since there are not so many distinct verbs that are unknown to the CCG lexicon, this is not so infeasible as it sounds.

(19) Unlike agonists for EP1 and EP3, agonists that bound EP2 or EP2 and EP4 receptors strongly inhibited expression of class II major histocompatability complex and CD23 and blocked enlargement of mouse B lymphocytes stimulated with IL-4 and/or lipopolysaccharide.
`=> inhibit(EP2+EP2&EP4 receptors,expression of class II major histocompatibility complex and CD23)` (WRONG)

The 'wrong subject' category also includes errors of the kind shown in (20). Here, the search for the NP on the left of *requires* failed to go beyond the underlined *by*, and produced the wrong relation. This can be fixed by adding the pattern to the module that proposes the NP candidates. In this case, the CCG parser simply did not even have the chance of looking into the correct NP candidate, as it was never proposed.

(20) Experiments employing inhibitors of cAMP metabolism demonstrate that the mechanism by which EP2 and EP4 receptors regulate B lymphocyte activity requires elevation of cAMP.
`=> activate(elevation,B lymphocyte activity)` (WRONG)

The 'wrong object' category is of a slightly different nature, but can be addressed similarly. Following the present suggestions and adding information about a few more syntactic structures, we predict that the precision rate can be raised to the range of 90 without making the system overly data-dependent. Although coordination is the construction where we lose valuable precision, it requires much heuristics (cf. Park and Cho[8]) to identify the intended pairing.

6 Conclusion

Given the task of extracting information regarding protein-protein interactions from natural language documents that show a heavily biased ratio of interaction types to protein names, we have chosen to jump into the verbs of interest and start looking for their syntactic arguments from there on, using a regular grammar to propose NP candidates liberally and a combinatory categorial grammar to validate them rigorously. The performance so far seems to be promising, though not quite satisfactory yet, considering the sensitivity of CCG to the natural language syntax that has not yet been brought to justice.

Future research directions include: identifying names of cells, proteins, and drugs, in the presence of many unknown words (cf. Rindflesch et. al.[2]) so that the extracted relations are further streamlined and related to one another in a more useful way; and rating the extracted relations according to various measures, such as authorship, mode of assertions, and the presence of additional qualifications to the findings.

Acknowledgments

We thank Limsoon Wong and anonymous reviewers for valuable comments.

References

1. C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: Protein-protein interactions. *Intelligent Systems for Molecular Biology.* 1999.
2. T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter. EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature. *Pacific Symposium on Biocomputing*, pages 517-528, 2000.
3. B. J. Stapley and G. Benoit. Biobibliometrics: Information Retrieval and Visualization from Co-occurrences of Gene Names in Medline Abstracts. *Pacific Symposium on Biocomputing*, pages 529-540, 2000.
4. J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS: Extracting Information from Natural Language Texts. In E. Roche and Y. Schabes, editors, *Finite State Devices for Natural Language Processing.* MIT Press, 1996.
5. J. Thomas, D. Milward, C. Ouzounis, S. Pulman and M. Carrol. Automatic Extraction of Protein Interactions from Scientific Abstracts. *Pacific Symposium on Biocomputing*, pages 541-552, 2000.
6. K. Humphreys, G. Demetriou, and R. Gaizauskas. Two Applications of Information Extraction to Biological Science Journal Articles: Enzyme Interactions and Protein Structures. *Pacific Symposium on Biocomputing*, pages 505-516, 2000.
7. M. Steedman. The Syntactic Process. The MIT Press, 2000.
8. J. C. Park and H. J. Cho. Informed Parsing for Coordination with Combinatory Categorial Grammar. *International Conference on Computational Linguistics*, pages 593-599, 2000.