Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser

Jing Ding Daniel Berleant
Department of Electrical and Computer Engineering, Iowa State University, Ames, IA
{dingjing, berleant}@iastate.edu

Jun Xu Andy W. Fulmer
The Procter and Gamble Company Cincinnati, OH
{xu.j.1, fulmer.aw}@pg.com

Abstract

Many natural language processing approaches at various complexity levels have been reported for extracting biochemical interactions from MEDLINE. While some algorithms using simple template matching are unable to deal with the complex syntactic structures, others exploiting sophisticated parsing techniques are hindered by greater computational cost. This study investigates link grammar parsing for extracting biochemical interactions. Link grammar parsing can handle many syntactic structures and is computationally relatively efficient. We experimented on a sample MEDLINE corpus. Although the parser was originally developed for conversational English and made many mistakes in parsing sentences from the biochemical domain, it nevertheless achieved better overall performance than a co-occurrence-only method. Customizing the parser for the biomedical domain is expected to improve its performance further.

1. Introduction

MEDLINE is a rich source for mining biochemical interactions for various tasks, such as populating databases of interacting proteins, constructing networks of protein interactions, and assisting human experts to sift through the most relevant documents. Many algorithms have been proposed in the literature, falling into two broad categories, statistical approaches and natural language processing (NLP) approaches. More attention has been paid to the latter, probably due to the fact that the single sentence (NLP's main focus) is a good choice of text unit for mining biochemical interactions [3], and many algorithms and tools can be borrowed from computational linguistics. In the existing reports, NLP approaches were used to analyze sentences with grammars of various levels of expressive power and computational complexity. However, it is challenging to find a good balance between the two factors. Most systems suffered from either not enough expressive power to deal with complex sentence structures, or too much computational overhead when processing complex sentences to be practical on large corpora. Yet MEDLINE abstracts are full of complex sentences, so without efficient handling of complex sentences, the overall performance and application of any NLP interaction mining system is limited.

In this paper, we propose extracting biochemical interactions from MEDLINE using a link grammar parser (LGP) [4]. The parser has good performance on complex sentences, attributable to its balance between expressive power and computational complexity. First, in section 2, we briefly discuss syntactic structures of MEDLINE sentences, and give a review of the strengths and limitations of the current NLP interaction mining approaches in the literature. In section 3 we give a brief introduction to link grammar and a link grammar parser. In section 4, we report the experiment results of using the parser on a sample MEDLINE corpus. Section 5 contains a discussion of the LGP's expressive power and computational complexity. Future developments and a conclusion follow.

2. Related work

Biochemical interactions described in MEDLINE abstracts are rarely stated as simply as "protein A activates protein B." Various syntactic structures are used to compact several interactions, as well as other information, into a single sentence. Among the most frequently used are nominalization (converting a predicate to a noun phrase) and coordination (combining two or more predicates with coordinating conjunctions). It is not uncommon in MEDLINE abstracts for a single sentence (or fragment) to describe a network of interactions. Consider an example from MEDLINE:

> *Gamma-aminobutyric acid mediation of the inhibitory effect of nitric oxide on the arginine vasopressin and oxytocin responses to insulin-induced hypoglycemia*. (PMID: 8952001)

This sentence fragment has four instances of nominalization and one of coordination. There are five biochemicals in the sentence (ten possible interacting pairs). Three interactions are explicitly described, NO → GABA, NO → AVP and NO → OXT), and four are implied (GABA

→ AVP, GABA → OXT, insulin → AVP and insulin → OXT (NO means nitric oxide, AVP means arginine vasopressin, OXT means oxytocin, and GABA means Gamma-aminobutyric acid. All are nominalized. Among the three non-interacting pairs, only AVP/OXT is obvious because of the coordination evidenced by the coordinating conjunction "*and*." The complexity shown here illustrates the challenges that NLP interaction mining algorithms face in this domain.

Natural Language Processing is still in its relative infancy, so automatic full discourse analysis and semantic understanding are beyond the capability of today's computing systems. Current NLP algorithms, therefore, try various ways to simplify the syntactic structures, or focus on specific subsets of the structures.

The syntactically simplest algorithms predict interaction from a pair of co-occurring terms, as in PathBinder [2]. Template matching algorithms require more evidence of interaction than just co-occurrence of the terms. One type of templates consists of two slots for interacting biochemicals and an interactor (an interaction-related word) slot, such as in [6] and [10]. Blaschke et al. [1] added a constraint that the interactor must be between the two terms. An implicit assumption behind such templates is that there is no crossover among predicates or nominalized predicates, so they performed poorly on multiple-instance nominalization and coordination. For example, a "term-interactor-term" template will miss two explicitly stated interactions in the example sentence given above (NO → AVP and NO → OXT), falsely include a non-interaction (GABA/insulin), either miss two implied interactions (insulin → AVP and insulin → OXT) or include a non-interaction (NO/insulin) depending on whether or not "*responses*" can fill the interactor slot. In addition, both "*mediation*" and "*inhibitory*" can fill the interactor slot between GABA and NO.

To better deal with nominalization, Leroy and Chen [5] tried to first find noun phrases using templates built around the preposition "*of*," and then fill the phrases into main templates built around the preposition "*by*." Other researchers took advantage of progress in the NLP field. For example commercial and open-source part-of-speech taggers and parsers were experimented with, such as in [7] and [11]. However, these approaches have some limitations. First, parsing is difficult, especially in face of the complexity of MEDLINE sentences. For example, Yakushiji et al. [11] experimented with a full parser on 179 sentences taken from MEDLINE. In only 66 (30%) were correct parse results obtained. Second, the parsing results (parse trees and grouped phrases) may or may not help extract interactions. Consider the example sentence mentioned above. The entire sentence (which is a title) is a noun phrase. Correct grouping of the entire phrase does not provide any useful information for interaction extraction. Park et al. [7] used a combinatory categorial grammar parser in their system to confirm their own noun phrase grouping rules, ignoring other output of the parser. In such systems, part of the computational resources goes into generating parse results that are irrelevant to the task at hand, which is a waste of computing resources. Finally, none of these algorithms dealt with coordination.

Coordination occurs in a sentence when it contains a shared structure. The sharing avoids duplication, so that the sentence is more compact than if sharing had not occurred. That is the main reason why this syntactic structure is so widely used in MEDLINE abstracts and elsewhere. Coordination can be applied to various sentence components. For example,

Protein A activates proteins *B and C.*
Protein A activates *protein B and protein C.*
Protein A *activates protein B, and inhibits protein C.*
…

All of these examples use coordination to avoid saying, for example, "*Protein A activates protein B. Protein A activates protein C.*" This kind of complexity cannot easily be handled by simple template matching. Note that the coordinating components (*B* and *C*) are not related to each other. They are put together only because they are related to a common third party. Therefore, coordination should be used to rule out interactions between the coordinating components.

In the next section, we introduce a link grammar parser, which deals with coordination. In addition, the output of the parser is also suitable for extracting relationships between non-coordinating terms.

3. Link grammar and the link grammar parser

Link grammar was first introduced by Sleator and Temperley to simplify English grammar with a context-free grammar [8]. The basic idea of link grammar is to connect pairs of words in a sentence with various links. Each word is viewed as a block with connectors coming out. There are various types of connectors, and connectors may point to the right or to the left. A valid sentence may have more than one complete linkage, just as a sentence may have several meanings.

Grinberg et al. [4] developed a robust parser to implement the link grammar. It has a dictionary of about 60,000 words, and can recognize a wide range of English syntactic phenomena: noun-verb agreement, questions, imperatives, complex and irregular verbs, many types of nouns, past- or present-participles in noun phrases, commas, a variety of adjective types, prepositions, adverbs, relative clauses, possessives, coordinating conjunctions, and others. The parser was tested on a corpus of English telephone conversations. Its robustness was demonstrated by its ability to handle many "ungrammatical" sentences and sentence fragments. If a complete linkage cannot be found,

Modified sentence: GABA mediates the inhibitory effect of NO on the AVP and OXT responses to insulin-induced hypoglycemia.

Parsing result (portion):

```
Linkage 6, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=30)
          +------------Os------------+           +----------Jp----------+
          |      +--------D*u--------+-----Mp----+  +--------Dmc--------+       +------------Jp-----------+
  +---Ss--+      |        +-----A----+--Mp-+Js+  |  |   +-------AN------+---Mp--+        +-------Ah-------+
  |       |      |        |          |     |  |  |  |   |               |       |        |                |
GABA mediates.v the inhibitory.a effect.n of NO on the AVP and OXT responses.n to insulin-induced hypoglycemia[?].n

          +------------Os------------+           +----------Jp----------+
          |      +--------D*u--------+-----Mp----+  +--------Dmc--------+       +------------Jp-----------+
  +---Ss--+      |        +-----A----+--Mp-+Js+  |  |           +---AN--+---Mp--+        +-------Ah-------+
  |       |      |        |          |     |  |  |  |           |       |       |        |                |
GABA mediates.v the inhibitory.a effect.n of NO on the AVP and OXT responses.n to insulin-induced hypoglycemia[?].n
```

Figure 1. A complete linkage (No. 6 among 50 total linkages) with two sub-linkages produced by the LGP on the sample sentence.

the parser will try to form a "partial linkage" by ignoring one or more of the words in the sentence. The parser has an internal timer. If the timer runs down before a complete or partial linkage has been found, the parser will output whatever it has found so far (termed a fragmented linkage).

The example sentence discussed in the previous section can serve as an example input for the LGP. To prevent the parser from making unnecessary mistakes, we abbreviated and capitalized the biochemical names. We also modified the sentence fragment slightly to make it a real sentence, but kept all the interactions unchanged. The modified sentence and one of the complete linkages are shown in Fig. 1. It took the parser 0.13 sec to process the sentence [9]. A total of 50 complete linkages were found. The parser has a cost system to express preferences among the linkages ("cost vector" in Fig. 1). For example, the parser may prefer the linkage with the shortest total link length. In this particular case, the linkage corresponding to the correct semantic meaning ranked 6th place. It has two sub-linkages (both shown in Fig. 1), because there was a coordinating conjunction in the sentence. The parser handles coordination by giving two sub-linkages. Each sub-linkage ignores one of the coordinating components (AVP and OXT). The parser also attached part-of-speech tags to some ambiguous words (noun, verb, adjective, etc.). The question mark following "*hypoglycemia*" meant that the word was not in the parser's dictionary, and was guessed to be a noun.

To find the interactions described in the sentence, we extracted the link paths between the ten pairs of terms (Table 1). For example, to extract the path between GABA and NO, we started at "`GABA`", followed the "`Ss`" link to "`mediates`", the "`Os`" link to "`effect`", the "`Mp`" to "`of`", and ended with the "`Js`" link to "`NO`" which constitutes four linking steps. The AVP/OXT pair can be excluded immediately from the interaction list because there is no link path between them. This is attributed to the parser's ability to handle coordinating conjunctions. In this particular case, the other two non-interacting pairs can also be excluded easily by a cut-off value for the number of links that must be traversed to get from one to the other (e.g. 6 or 7 in this case). In general, it may not be clear what such a cutoff value should be. However, even if these two pairs were not excluded, we already have a gain in precision without any loss in recall compared to taking co-occurrence alone as evidence of interaction.

Note the tendency of the words in the link paths to be relevant to the chemical pairs of interest. These fragments could be further processed using methods such as those reviewed in the previous section.

4. Experiment results on the IEPA corpus

In order to see how the LGP works in a MEDLINE set-

Table 1. Extracted link paths between biochemical pairs from the LGP's output.

Pair	Relevant fragment (link path)	Steps	Interaction
GABA→NO	GABA mediates effect of NO	4	Explicit
NO→AVP	effect of NO on AVP response	5	Explicit
NO→OXT	effect of NO on OXT response	5	Explicit
GABA→AVP	GABA mediates effect on AVP response	5	Implied
GABA→OXT	GABA mediates effect on OXT response	5	Implied
AVP→insulin	AVP response to insulin-induced hypoglycemia	4	Implied
OXT→insulin	OXT response to insulin-induced hypoglycemia	4	Implied
GABA→insulin	GABA mediates effect on response to insulin-induced hypoglycemia	7	None
NO→insulin	effect of NO on response to insulin-induced hypoglycemia	7	None
AVP→OXT	(None)		None

ting, we experimented with it on our Interaction Extraction Performance Assessment (IEPA) corpus [3]. This corpus has approximately 485 sentences taken from 303 abstracts. Each sentence contains at least one pair of biochemicals of interest. In order to increase parsing efficiency and accuracy, the sentences were manually preprocessed using the following rules:

1. Capitalize the chemical names of interest in the sentence.
2. Connect compound names with an underscore.
3. Replace special characters such as /, +,), (, and ' within chemical names with underscore,.
4. Simplify the sentence as follows:
   If the chemical pair occurs within a sentence fragment between two consecutive punctuation marks in {`,` `;` `.`}, manually feed the fragment to the parser. If a complete linkage can be found, or link paths between the pair can be extracted from a partial or fragmented linkage, use the fragment only. Otherwise, use the entire sentence.
5. Modify a title into a normal sentence as follows:
   - De-capitalize each word except the chemical names.
   - Apply rule 4, if applicable.
   - For a fragmental title or a two-sentence title:
     if no complete linkage is found, nor can any link path be extracted from a partial or fragmented linkage, and the sentence matches either the pattern *string: string2* or the pattern *string1. string2*, change the sentence to "*string1 (string2) is described*".
     Otherwise append "*is described*".

A Java program (available upon request) then fed the modified sentences to the parser, read the output, and extracted link paths between the two biochemicals. A sentence may have more than one pair of biochemicals; in that case paths were extracted for each pair. For sentences with multiple linkages, the first-ranked one, according to the parser's default cost ranking system, was used to extract link paths, regardless of whether or not it was semantically correct. Finally, the results were compared with manual analyses of the sentences.

Although we automated some steps in the experimental procedure, we did not build a complete interaction mining system. The focus of this study was the capability and performance of the LGP on interaction extraction, not on constructing a complete system. If the parser were tested as part of a complete system, the performance of other modules (e.g. a biochemical name recognizer) would make it harder to interpret the results. However, the manual processing steps were specifically designed to be easily coded in software and to avoid requiring human judgment. For example, many titles are sentence fragment(s), such as the example in the previous section. Another example is "*Anorexia nervosa and bulimia nervosa: An appraisal*" (PMID: 12768223). To prevent the parser from wasting time and making errors on attempting to construct complete linkages from such structures, we assumed them to be noun phrases and converted them to sentences by appending the phrase "*is described*" (rule #5).

Out of 644 co-occurring biochemical pairs in the IEPA corpus, the parser found link paths between 429 pairs (timeout: 15 sec) or 476 pairs (timeout: 10 min), as summarized in Table 2. A closer look at the extracted link paths, as well as the parser's original output, revealed that the parser made a considerable number of mistakes. The mistakes may be categorized into four groups:

1. *Unknown words*. The parser was targeted at conversational English. It did not recognize many words in MEDLINE abstracts. Although it made a correct guess in the example sentence in the previous section, the chances of correctly guessing multiple unknown words decreases rapidly. In addition, the parser did not use biomedical domain knowledge in its guessing rules (e.g. a word ending with "-ase" is very likely to be an enzyme name, a noun).
2. *Unfamiliar structures*. The parser was confused by some structures frequently seen in MEDLINE abstracts, such as probabilities (*R2=0.170*, *p<0.0001*), comparisons (*chemical 1 > chemical 2 > chemical 3*) and units (*mg/d*, etc.).
3. *Ranking*. The semantically correct linkage may not be ranked first by the parser's default cost system, as shown in the example sentence in the previous section.
4. *Sentences that were too long and complicated*. All too often the internal timer ran down, and the parser then returned only fragmented linkages.

Table 2. Result of the LGP on the IEPA corpus.

	Timeout: 15 sec				Timeout: 10 min			
	Interaction type			Sub-total	Interaction type			Sub-total
	Ex	Im	None		Ex	Im	None	
Link path found	210	70	149	429	215	77	184	476
No link path found	29	27	159	215	24	20	124	168
Subtotal	239	97	308	644	239	97	308	644

Ex=explicit; Im=implied.

Since the extracted link paths between a biochemical pair were probably incorrect, we did not try further analyses on the exact links, such as setting a cut-off value on the number of link steps or developing templates to filter link types. Instead, we only used the existence or not of a link path between a pair as a decision rule: no link path, no interaction. Two reasons may cause the lack of a link path between a chemical pair. First, the pair is connected with a coordinating conjunction, as illustrated in the example sentence. This is a case where rejection is exactly what one hopes for. The other reason is that a sentence might be too complicated and only a fragmented linkage is found for it. This type of rejection can be interpreted as suggest-

ing that the syntactic connection between the two terms is tenuous enough that there is little chance of interaction between them. As shown in Table 2, when no link path was found, there was a relatively higher chance that there was in fact no interaction between the terms. Compared to co-occurrence of two biochemical names as an indicator of an interaction between them, precision improved 25% (13 percentage points), outweighing the loss in recall and resulting in a modest net increase of information retrieval effectiveness of 4–5% (Table 3). The recalls achieved using the decision rule were, not surprisingly, higher for explicitly stated interactions, as the table shows. Giving the parser more time (e.g. with a timeout of 10 min) did not lead to better results (the information retrieval effectiveness score went from 0.73 to 0.72).

Table 3. Performance of the link decision rule.

	Co-occur. only [3]	Link rule (15 sec)	Link rule (10 min)
Recall (explicit)	100%	88%	90%
Recall (implied)	100%	72%	79%
Recall (overall)	100%	83%	87%
Precision	52%	65%	61%
Effectiveness	0.68	0.73	0.72

5. Discussion

The LGP's ability to handle coordinating conjunctions and other English syntactic phenomena is attributable to its expressive power as a context-free grammar [8]. Context-free grammars are more powerful than regular expressions. Will algorithms using more powerful grammars, such as context-sensitive or free grammar, be better for interaction extraction than the LGP? Potentially this must be the case if human NLP capabilities are assumed to be automatable at speeds faster than humans are capable of. However this is not on the horizon. Currently it is not necessarily the case that more powerful grammars lead to better biochemical interaction extraction.

Context-free grammars have polynomial complexity. In the case of the LGP, the worst-case complexity is cubic [4]. This is important as polynomial algorithms are considered tractable, while exponential algorithms are considered intractable in the sense that as problem size increases, the computation required to solve it rapidly becomes unavailable.

Although we have demonstrated that the LGP has the potential to be a useful part of a system for extracting biochemical interactions, its current limitations are also evident, as highlighted by the moderate performance gain in our experiment. Below is a list of further developments that would enhance the value of link grammar parsing in the biomedical domain.

- Extend its dictionary to include technical terms.
- Extend its unknown-word-guessing rules, so that, for example, the parser can guess that a word ending with "`-ase`" is a protein name and not a verb.
- Fine-tune its cost system for better ranking.
- Develop other algorithms, such as template matching, to further process link paths extracted from the parser's output.

In conclusion, the LGP showed that link grammar parsing can be a useful tool for interaction mining from MEDLINE, particularly from the standpoint of precision. Its value could be enhanced by customizing it to the biomedical domain.

6. References

[1] Blaschke, C., M. Andrade, C. Ouzounis, and A. Valencia (1999) Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions. *AAAI Conference on Intelligent Systems in Molecular Biology 60-67.*

[2] Ding, J. (2003) PathBinder: A Sentence Repository of Biochemical Interactions Extracted from MEDLINE (MS thesis). Iowa State University, Ames, IA.

[3] Ding, J., D. Berleant, D. Nettleton, and E. Wurtele (2002) Mining MEDLINE: Abstracts, Sentences, or Phrases? *Pacific Symposium on Biocomputing* 7: 326-337.

[4] Grinberg, D., J. Lafferty, and D. Sleator (1995) A Robust Parsing Algorithm for Link Grammars. *Proceedings of the Fourth International Workshop on Parsing Technologies.*

[5] Leroy, G. and H. Chen (2002) Filling Preposition-Based Templates to Capture Information from Medical Abstracts. *Pacific Symposium on Biocomputing* 7: 350-361.

[6] Ng, S.-K. and M. Wong (1999) Toward Routine Automatic Pathway Discovery from On-Line Scientific Text Abstracts. *Genome Informatics* 10: 104-112.

[7] Park, J.C., H.S. Kim, and J.J. Kim (2001) Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorial Grammar. *Pacific Symposium on Biocomputing* 6: 396-407.

[8] Sleator, D. and D. Temperley (1993) Parsing English with a Link Grammar. *Third International Workshop on Parsing Technologies.*

[9] Temperley, D., D. Sleator, and J. Lafferty Link Grammar Parser Online Demo. http://www.link.cs.cmu.edu/link/submit-sentence-4.html

[10] Wong, L. (2001) PIES, a Protein Interaction Extraction System. *Pacific Symposium on Biocomputing* 6.

[11] Yakushiji, A., Y. Tateisi, Y. Miyao, and J. Tsujii (2001) Event Extraction from Biomedical Papers Using a Full Parser. *Pacific Symposium on Biocomputing* 6: 408-419.