BIOINFORMATICS

Vol. 19 no. 16 2003, pages 2046–2053
DOI: 10.1093/bioinformatics/btg279

Extraction of protein interaction information from unstructured text using a context-free grammar

*Joshua M. Temkin and Mark R. Gilder**

GE Global Research, 1 Research Circle, Niskayuna, NY 12309, USA

Received on February 18, 2003; revised on April 14, 2003; accepted on April 26, 2003

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

ABSTRACT

Motivation: As research into disease pathology and cellular function continues to generate vast amounts of data pertaining to protein, gene and small molecule (PGSM) interactions, there exists a critical need to capture these results in structured formats allowing for computational analysis. Although many efforts have been made to create databases that store this information in computer readable form, populating these sources largely requires a manual process of interpreting and extracting interaction relationships from the biological research literature. Being able to efficiently and accurately automate the extraction of interactions from unstructured text, would greatly improve the content of these databases and provide a method for managing the continued growth of new literature being published.

Results: In this paper, we describe a system for extracting PGSM interactions from unstructured text. By utilizing a lexical analyzer and context free grammar (CFG), we demonstrate that efficient parsers can be constructed for extracting these relationships from natural language with high rates of recall and precision. Our results show that this technique achieved a recall rate of 83.5% and a precision rate of 93.1% for recognizing PGSM names and a recall rate of 63.9% and a precision rate of 70.2% for extracting interactions between these entities. In contrast to other published techniques, the use of a CFG significantly reduces the complexities of natural language processing by focusing on domain specific structure as opposed to analyzing the semantics of a given language. Additionally, our approach provides a level of abstraction for adding new rules for extracting other types of biological relationships beyond PGSM relationships.

Availability: The program and corpus are available by request from the authors.

Contact: gilder@research.ge.com; jtemkin1@comcast.net

INTRODUCTION

The recent publication of the Human Genome Draft Sequence (Lander *et al.*, 2001; Venter *et al.*, 2001) and the rapid proliferation of data generated from biological assays such as Yeast-2-Hybrid, Co-immunoprecipitation and Microarray analysis, continues to accelerate the rate at which new discoveries are being published. These discoveries often contain novel observations about proteins, genes, and small molecules (PGSMs) such as pharmaceuticals and other foreign compounds and their roles in disease pathology and cellular function. Several efforts such as BIND (Bader *et al.*, 2001), KEGG (Kanehisa *et al.*, 2002), EcoCyc (Karp *et al.*, 2002), DIP (Xenarios *et al.*, 2002), MINT (Zanzoni *et al.*, 2002), and Transpath (Schacherer *et al.*, 2001) have been developed to store these data in structured formats allowing for the production of protein interaction maps. However, these databases remain sparsely populated requiring manual curation and interpretation of the literature in order to populate them with new experimental data pertaining to PGSM interactions. This problem continues to accelerate as the rate of new publications containing PGSM interactions continues to grow. In order to improve the process of populating these data sources, new techniques and algorithms capable of accurately and efficiently extracting interaction data from the vast corpus of scientific literature are required.

Research in the fields of Information Extraction and Natural Language Processing (NLP) has been focused on developing techniques to overcome the highly ambiguous and variable nature of natural language in order to extract information from unstructured text. Direct application of this research work to the area of biological text extraction has been focused on the accurate and efficient recognition and classification of PGSM names and the extraction of their biological interactions.

Various techniques for recognizing PGSM names have been proposed. For instance, the use of standardized dictionaries containing the names and synonyms of PGSM has been shown to be an effective way for recognizing these entities in free form text (Blaschke *et al.*, 1999; Rindflesch *et al.*, 1999). Although applications of this technique have reported high rates of recall and precision, this technique remains limited as PGSM names not present in the dictionaries produce large amounts of false negatives. Others have addressed the issue of false negatives by using

*To whom correspondence should be addressed.

Bioinformatics 19(16) © Oxford University Press 2003; all rights reserved.

templates capable of recognizing common naming patterns for PGSMs (Fukuda *et al.*, 1998; Ng and Wong, 1999; Yu *et al.*, 2002). These techniques, which scan potential names by looking for patterns of capitalization, numbering, and use of hyphens have been shown to capture many of the entities missed by the dictionary approach alone, thereby reducing the amount of false negatives. However, these techniques have also been shown to generate a large number of false positives by recognizing words that match the templates but are in fact not PGSM names. Alternative approaches have addressed the problems of name recognition through the use of machine learning (Proux *et al.*, 1998; Hatzivassiloglou *et al.*, 2001), and through the use of statistics (Krauthammer *et al.*, 2000; Tanabe and Wilbur, 2002). Although these techniques have reported incremental gains in overall recall and precision over the template and dictionary based approaches, it has been shown that these techniques are also limited by the quality and extent of the training sets used to train the algorithms (Tanabe and Wilbur, 2002).

Similar to the problem of identifying PGSM names, there has been a wide range of varying techniques published for extracting relationships from scientific literature. For example, several have shown that template and simple rule based algorithms can be used to recognize interactions achieving high rates of recall and precision (Blaschke *et al.*, 1999; Ng and Wong, 1999; Ono *et al.*, 2001; Wong, 2001; Pustejovsky *et al.*, 2002). However, this technique has been found to be overall limited in the set of interactions that can be extracted by the extent of the recognition rules that are implemented, and also by the complexity of sentences being processed. Specifically, complicated cases such as interaction descriptions that span several sentences of text are often missed by these approaches. Others have addressed the issue of complex sentence structures and some limited work has been done on extracting interactions spanning several sentences through the use of parts of speech analysis (Humphreys *et al.*, 2000), and natural language based approaches (Rindflesch *et al.*, 2000; Friedman *et al.*, 2001). These approaches, like the rule-based systems, have also reported high levels of recall and precision. However, as noted in a recent review article by Hirschman *et al.* (2002), a lack of a standard common corpus and a lack of standard techniques and equations for reporting recall and precision, has made comparative analysis of different techniques a difficult problem.

In this paper, we describe an alternative method for extracting PGSM interactions from natural language that achieves high rates of recall and precision using a lexical analyzer and an extensible context-free grammar (CFG). We address the difficulty of natural language processing, by filtering the input text into a stream of tokens and using an extendable CFG designed specifically for parsing biological text. As we will show, CFG provide an easily extendible platform for extracting interactions from free text and are powerful enough to describe most natural language structure while being able to be restricted enough to allow for efficient parsing. We also describe a methodology for creating a corpus for analyzing techniques that can be extended and potentially used to do comparative analysis between techniques in the future.

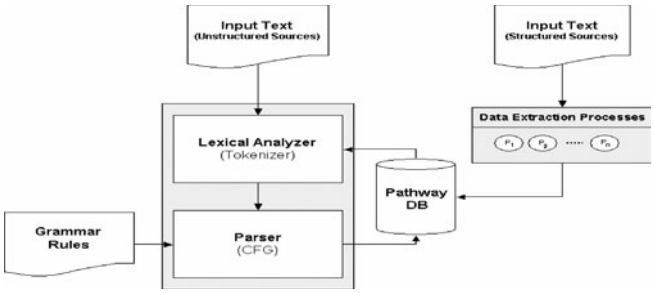

Fig. 1. System topology.

SYSTEM AND METHODS

Overview

Our method for extracting PGSM interactions from unstructured texts can be divided into three separate parts:

(1) a Pathway Database (PDB) consisting of dictionaries that are used by

(2) a Lexical Analyzer to tokenize and tag relevant terms from scientific abstracts retrieved from PubMed whose output stream of tokens is then passed to

(3) a Parser constructed around a CFG that is used to interpret the collection of tokens and output interactions based on the rules of the grammar (Fig. 1).

We designed and built the system using the Java (TM Sun Microsystems) programming language, and utilized the JavaCC (WebGain) compiler to generate the CFG.

Pathway database dictionaries

The PDB consists of two distinct dictionaries: (1) a name dictionary for recognizing PGSM names and their synonyms, and (2) a category/keyword dictionary for identifying terms describing interactions. The name dictionary was constructed by combining a limited set of PGSM names from Swissprot (Bairoch and Apweiler, 2000), GenBank (Benson *et al.*, 2002), and KEGG (Kanehisa *et al.*, 2002). The resulting name dictionary consists of 67 326 unique names and synonyms describing a total of 37 546 distinct entities. The category/keyword dictionary was adapted from Friedman *et al.* (2001) and the NIH relevant term list for oncogene expression (NIH, 1999) with additional categories and keywords found to be prevalent in our corpus as shown in Table 1.

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

Table 1. Interaction keywords

Category	Keywords	Category	Keywords	Category	Keywords
Activate	accumulat (e,ed,es,ion)	Break bond	cleav (e,ed,es)	Inactivate	inhibit (s,ed,ion)
	activat (e,ed,es,or, ion)		demethylat (e,ed,es,ation)		reduc (e,ed,es,tion)
	elevat (e,ed,es,ion)		Dephosphorylat (e,ed,es,ation)		repress (ed,es,ion)
	hasten (ed,es)		sever (e,ed,es)		supress (ed,es,ion)
	Incite (ed,es)	Cause	influenc (e,ed,es)	Modify	modifi (ed,cation)
	increas (ed,es)	Contain	contain (s,ed,es)	Process	apoptosis
	Induc (e,ed,es,tion)	Create bond	methylat (e,ed,es,ation)		myogenesis
	Initiat (e,ed,es,ion)		phosphorylat (e,ed,es,ation)	React	interact (s,ed,ion)
	promot (e,ed,es)	Generate	express (ed,es,ion)		react (s,ed,ion)
	stimulat (e,ed,or,ion)		overexpress (ed,es,ion)	Release	disassembl (e,es,ed)
	transactivat (e,ed,es,ion)		produc (e,ed,es,tion)		discharg (e,es,ed)
	up-regulat (e,ed,es,or,ion)	Inactivate	block (s,ed)	Signal	mediat (e,ed,es)
	Upregulat (e,ed,es,or)		decreas (e,ed,es)		modulat (e,ed,es,ion)
Association	associat (e,ed,es,ion)		deplet (e,ed,es,ion)		participat (e,ed,es,ion)
Attach	add (s,ition)		down-regulat (e,ed,es,ion)		regulat (e,es,ed,ion)
	bind (s),bound		downregulat (e,ed,es,ion)	Substitute	replac (e,ed,es)
	catalyz (e,ed,es)		impair (s,ed)		substitut (e,ed,es,ion)
	Complex		inactivat (e,ed,es,ion)		

Lexical analyzer

The lexical analyzer was designed to accept both unstructured text in addition to PubMed abstracts from the web. It then parses the input and generates a stream of tagged tokens based on the descriptions in Table 2.

The lexical analyzer tags the input text by iterating through the document as shown in Figure 2. The initial step of the process involves the identification and delimitation of sentence boundaries. Each step beyond this initial process utilizes the dictionaries in the pathway database for word recognition and tagging. We have adapted the set of protein and gene name recognition rules described by Fukuda *et al.* (1998) in order to limit the occurrence of false negatives for names that the lexical analyzer does not recognize during the tagging of the input text. Only words that match those stored in the dictionaries or those that match based on the adapted name recognition rules are converted to tokens and placed in the output stream.

The resulting output stream of tokens is available for the parsing phase of the overall process. This phase is responsible for analyzing the token stream using the set of CFG productions for the purposes of extracting interaction information. As illustrated in Figure 2, the lexical analyzer and parser are separate component processes that communicate via the token stream allowing other third-party tools to be easily integrated. As previously mentioned we chose a Java implementation because of its ease of portability to all major system platforms.

Context-free grammar

The parser was developed using a concise set of grammar production rules allowing for the detection of PGSM interactions. The production rules were derived by manually analyzing a large corpus of 500 non-topic specific scientific abstracts pulled from PubMed containing various representations of interaction data in unstructured text. Biochemists read and highlighted the abstracts for relevant sentences describing interactions that were then used to derive the production rules. The resulting production rules were combined and represented in a CFG.

The use of CFGs for validating structure in natural language was first proposed by Chomsky (1956). A CFG for representing production rules has four key components as described by Aho *et al.* (1986).

1. A set of tokens T, known as terminal symbols.
2. A set of non-terminals N disjoint from T.
3. A set of productions P of the form $a \rightarrow b$, where $a \in N$ and b is a sequence of one or more symbols from $N \cup T$.
4. The start symbol S where $S \in N$.

Therefore, the language generated by a CFG can be enumerated by repeatedly applying production rules, commencing with the start symbol S, and replacing non-terminals with their associated production rules until all non-terminals have been processed.

To address the problem of extracting PGSM interactions from unstructured text we developed the grammar illustrated in Table 3 using EBNF (Extended Backus–Naur Form).

The representation of the grammar described in Table 3 depicts terminal symbols using **bold** print and non-terminals using plain text. A description for each non-terminal symbol is provided in Table 4.

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

Table 2. Tags recognized and produced by lexical analyzer

Lexical analyzer tags	Description	Example
BETWEEN	Tag for the word 'between'	Complex **between** *x*, *y*, and *z*
BY	Tag for the word 'by'	*X* blocked **by** *Y*
EOC	Tag to indicate logical end-of-sentence	
KEY	Tag for interaction keywords stored in the pathway database (Fig. 2)	see Figure 2
MOL	Tag for combination of long entity names with their associated abbreviated names	Interleukin 10 (IL-10)
MOL_LONG	Tag for recognized sets of words which match long name descriptions in the DB	Interleukin 10
MOL_SHORT	Tag for recognized abbreviation or short name forms of entities that match entity names stored in the pathway database	IL-10
NEGATOR	Tag for words or sets of words which negate the sentence	*X* **did not** inhibit *y* *X* **was not** shown to inhibit *Y* *X* inhibited *Y*, *Z*, *Q*, **but not** *R*
OF/ON	Tag for 'of' and 'on' in a sentence	Addition **of** *X* **on** *Y*
OF/BY	Tag for 'of' and 'by' in a sentence	Inhibition **of** *X* **by** *Y*
TRANSITIVE	Tag for the word 'and' when followed by a keyword	*X* blocks *Y* **and** upregulates *Z*
WITH	Tag for the word 'with'	*X* blocks the expression of *Y* **with** *Z*

As previously stated, tokens are supplied to the parser by the lexical analyzer. The parser utilizes the non-terminals to validate that the input stream of tokens is part of the language described by the grammar.

In general parsing methods fall into one of the two categories, top-down and bottom-up. In top-down parsing, also known as recursive descent, construction starts at the root nodes and progresses towards the leaves while in bottom-up parsing construction starts at the leaves and progresses upward. The JavaCC tool from WebGain generates a top-down LL(1) parser, which allows the use of more general grammars although left recursion is disallowed. As noted by (Aho

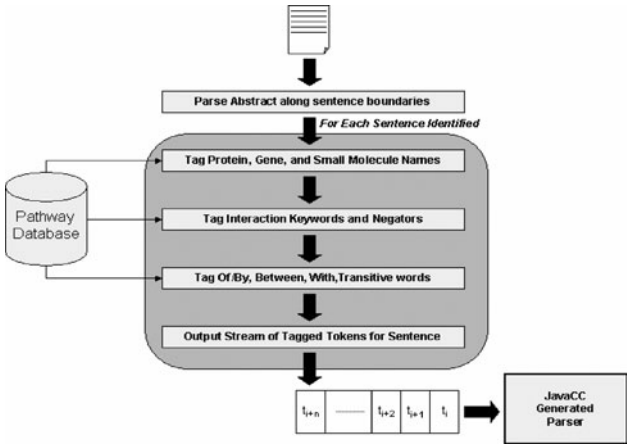

Fig. 2. Lexical analyzer processing.

Table 3. Grammar in EBNF form

S (start symbol)	Interactions
Interactions	MolExpr Interactions | MolExpr
MolExpr	Assignment | Relationship
Assignment	Expr (Negator)? **KEY** (Relationship_Conj)? Expr (**TRANSITIVE KEY** expr)* Eoc
Eoc	**EOC**
Relationship	**KEY** Relationship_Conj Expr (Relationship_Prep Relationship_Obj)? (**KEY**)* Eoc
Relationship_Conj	**BETWEEN** | **OF**
Relationship_Prep	**BY** | **ON** | **WITH**
Relationship_Obj	Expr
Expr	(Negator)? Molecule ((Negator?) Molecule)*
Negator	**NEGATOR**
Molecule	**MOL** | **MOL_LONG** | **MOL_SHORT**

et al., 1986), trade-offs exist between these two approaches. However, the use of a top-down parser for this application does not impair the overall performance or effectiveness of using this technique for extracting interaction data.

Example

An example of the lexical analyzer and parser working in tandem to tag, validate, and parse an input sequence is illustrated in Figure 3.

A sample sentence from the corpus is presented Figure 3a. Each word of the input text is read by the lexical analyzer and checked against the PDB for matches. If a match is found, the lexical analyzer builds the corresponding tagged token containing the original source text and passes it to the JavaCC generated parser (Fig. 3b). The parser validates the token sequence by applying the grammar productions, beginning with the start symbol *S*, consuming the stream of tokens generated by the lexical analyzer as illustrated by the parse tree

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

Table 4. Non-terminal symbol descriptions

Non-terminal symbol	Description
S	Defines 'Start' symbol used to identify starting point for of the grammar
Interactions	Defines possible Interaction types
MolExpr	Defines types of molecule expressions
Assignment	Defines assignment expression
Relationship	Defines relationship expression
Expr	Defines general form for expressions
Negator	Maps directly to terminal symbol NEGATOR
Relationship_Conj	Maps to one of the following terminal symbols: BETWEEN or OF
Relationship_Prep	Maps to one of the following terminal symbols: BY, ON or WITH
Relationship_Obj	Maps directly to the non-terminal symbol: Expr
Molecule	Maps to one of the following terminal symbols: MOL, MOL_LONG or MOL_SHORT
Eoc	Maps directly to terminal symbol EOC
ε	Indicates an epsilon production, i.e. the empty string
?	Indicates 0 or 1 occurrences of the specified quantity, i.e. optional
*	Indicates 0 or more occurrences of the specified quantity

in Figure 3c. In this particular case, the application of several production rules results in the generation of an empty string, denoted by an epsilon. The net effect of applying ε-productions is to remove the non-terminal from the string being generated as shown. We note that a common feature of parser–generator tools, like JavaCC, is the ability to separate application specific logic from the parsing rules. This is illustrated by the recognition of the *MolExpr* production rule shown in Figure 3d, where the interaction is inserted into the database.

Figure 4a–c illustrates the addition of negative interactions that are initiated by encountering the NEGATOR token derived from the text: *did not*. Figure 4d–f illustrates the production of two false negatives by the failure of the lexical analyzer to recognize the E-selectin and P-selectin proteins. Figure 4g–i illustrates the generation of a large number of false positives when the lexical analyzer and CFG encounter sentences with a problematic sentence structure.

RESULTS AND DISCUSSION

In order to test our lexical analyzer and CFG, we developed a test corpus from 100 randomly selected scientific abstracts from PubMed and were not part of the original 500 abstracts used to derive the production rules. The corpus was manually analyzed for PGSM names in addition to any interaction relationships present in each abstract within the corpus by biochemists within our laboratory. Analysis of the corpus revealed 562 distinct references to PGSM names and a total of 239 distinct references to interaction relationships. Tests of the system were performed over the same set of 100 articles, by capturing the set of molecules and interactions recognized by the system and comparing this output against the manually analyzed results previously described. We measured recall and precision rates for both the ability to recognize PGSM names in text in addition to the ability of the system to extract interactions based on the following calculations:

1. Recall = TP/(TP + FN)
2. Precision = TP/(TP + FP)

where TP, FN, and FP are defined as:

- TP—is the number of PGSM names; or interactions that were correctly identified by the system and were found in the corpus.
- FN—is the number of PGSM names; or interactions that the system failed to recognize in the corpus.
- FP—is the number of PGSM names; or interactions that were recognized by the system but were not found in the corpus.

Analysis of the output generated by the system demonstrated recall and precision rates for recognizing PGSM names to be 83.5 and 93.1%, respectively, while the recall and precision rates for extracting interactions was calculated to be 63.9 and 70.2%, respectively.

The results show that the system performs accurate extractions of interaction data when the lexical analyzer and parser encounter sentences that match the specified grammar. This includes complex sentences such as the one depicted in Figure 3, where the method correctly identified and extracted the observation that endothelin is involved in the activation of ERK and p38 MAP Kinase. Additionally, the system accurately extracts observations that dispute potential interactions such as those depicted in Figure 4a–c, which is an example of the system correctly extracting the fact that Ox-LDL did not change the expression of FASL, FADD or FLICE.

For the cases in which the system generated false negatives and/or false positives while extracting interactions, the root cause was typically due to one of the following conditions regarding the lexical analyzer:

1. The lexical analyzer failed to correctly identify a word as a PGSM leading to generation of a false negative, or
2. The lexical analyzer incorrectly identified a word as a PGSM leading to generation of a false positive.

An example of the first condition is shown in Figure 4d–f where the failure of the lexical analyzer to correctly recognize the E-Selectin and P-Selectin protein names resulted in the subsequent failure of the CFG rules to extract the interaction

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

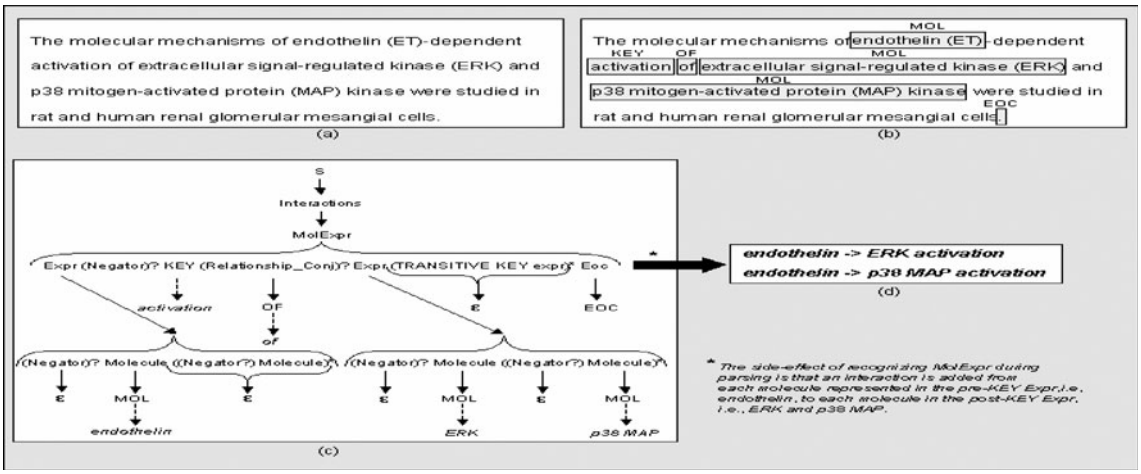

Fig. 3. Lexical analysis and parsing example.

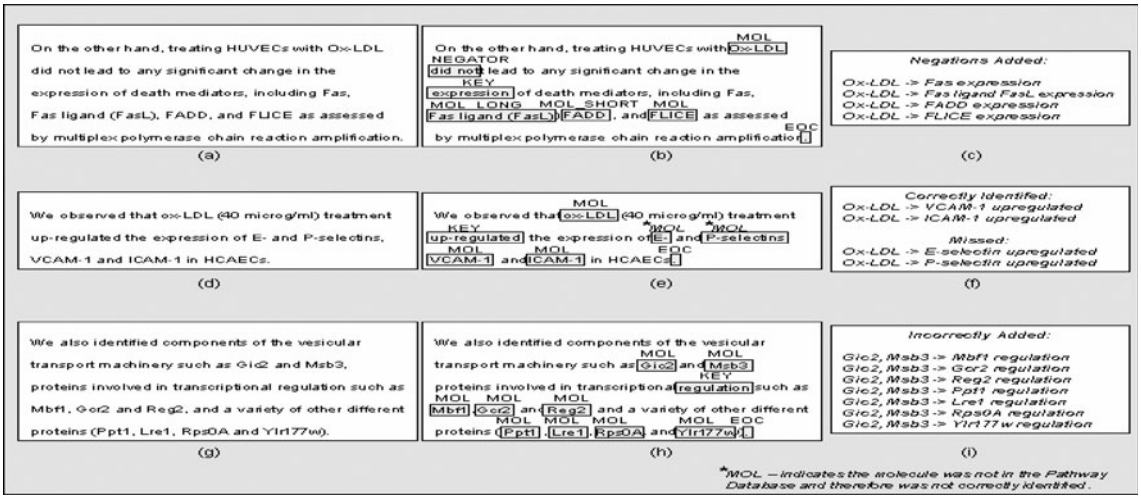

Fig. 4. Additional examples.

between Ox-LDL and these entities. However, we anticipate that through expanding the content of the PGSM dictionary this problem can be obverted and subsequently eliminate many of the false negatives generated by this condition raising overall recall and precision of the system.

Additionally, the results demonstrate that the lexical analyzer and grammar have some limitations in recognizing interactions and can generate false positives when sentences such as the one described in Figure 4g–i are encountered. In this case, a long sentence describing three separate ideas was encountered. Analysis of this sentence revealed that commas and no other logical indicators were present to separate the three distinct ideas being conveyed. This resulting output from the lexical analyzer was a valid stream of tokens that matched a set of rules within the grammar leading to the creation of a large number of false positive interactions. If the boundaries separating these ideas had been correctly identified these additions could have been avoided. However, this example does demonstrate how various unstructured text representations can negatively impact overall effectiveness. We note that in general, the incidence of false-positive entries were minimal as shown by the high level of precision rates achieved from the overall analysis of the corpus.

Direct comparison of our work to the work of others is not possible, since the corpuses used to evaluate and report on each method have differed due to the lack of a standard corpus in the field. Despite these limitations, we have shown that our approach generates high recall and precision rates using a

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

broad, non-topic specific corpus, whereas other reported results have focused on smaller sets of pre-selected articles or specific domain topics. For example, in the area of recognizing names in text, many have reported high level of recall and precision using varying corpuses. Fukuda *et al*. has reported a recall of 99% and precision of 95% for recognizing protein names in text. However, the corpus used consisted of 30 abstracts specific to the SH3 domain in addition to 50 abstracts specific to the topic of signal transduction (Fukuda *et al.*, 1998). Proux *et al*. has reported a recall of 94% and precision of 91% using a corpus of 1200 sentences containing gene names found within the Drosophila genome (Proux *et al.*, 1998) while Krauthammer *et al*. has reported a recall of 79% and precision of 72% compiled using a single hand curated review article (Krauthammer *et al.*, 2000).

Similarly, when comparing techniques used to extract interactions from text, corpuses have also varied. For example, Friedman *et al*. has reported a recall of 63% and precision of 96%. However, these results were compiled from a single hand-annotated paper from Cell (Friedman *et al.*, 2001). Ono *et al*. has reported a recall rate between 83 and 86% and precision of 94% using a corpus consisting of a collection of sentences specific to the Yeast and *Escherichia coli* genomes where each sentence within the corpus had at least two protein names and at least one interaction keyword present in the text (Ono *et al.*, 2001). Pustejovsky *et al*. reported a recall of 57% and precision of 90% using a corpus of 500 hand annotated abstracts all having the property of having derivatives of the word inhibit present within the text (Pustejovsky *et al.*, 2002).

Our results do show that our reported recall and precision rates for each of these problems fall within these reported ranges despite the use of different corpuses. However, our use of a broad based non-species or topic specific corpus for analyzing recall and precision may offer a more representative measure for reporting the overall effectiveness of an extraction technique for generalizing what level of accuracy can be expected when the technique is applied over the entire set of available scientific literature. We therefore conclude that our reported recall and precision rates for extracting PGSM interactions from unstructured text shows the potential to be able to mine the larger set of scientific literature available in order to populate structured representations for capturing interaction data for further computational analysis.

CONCLUSION

In conclusion, we have demonstrated that the problem of extracting PGSM interactions from unstructured text can be solved with high rates of recall and precision by using a CFG to recognize specific patterns used to describe interactions. We have shown that the use of a lexical analyzer and domain specific dictionary to convert the unstructured text into a stream of parsable tokens effectively reduces the problem of information extraction into one of pattern matching that can be efficiently solved by CFGs. This approach significantly reduces the complexities associated with natural language processing. We have also demonstrated that the addition of new rules can be easily facilitated by the use of CFGs. We intend on further utilizing this feature to expand the capabilities of the system to be able to extract other relationships such as reported correlations between the expression of certain genes and proteins in disease processes, tissues and cells. In addition, our approach of using a non-topic specific corpus can be used to analyze the overall accuracy of an extraction technique for achieving the overall goal for mining the larger set of scientific literature available in the field.

ACKNOWLEDGEMENTS

We would like to thank Brion Sarachan and Nadeem Ishaque of GE Global Research and Eric Stahre of GE Medical Systems for their continued support and funding for this research. In addition, we would like to thank our colleague Melvin Simmons also of GE Global Research for his insights and suggestions in building and refining the grammar and corpus used in this study.

REFERENCES

Aho,A.V., Sethi,R. and Ullman,J.D. (1986) *Compilers: Principles, Techniques, and Tools.* Addison-Wesley, Reading, MA.

Bader,G.D., Donaldson,I., Wolting,C., Ouellette,B.F., Pawson,T. *et al.* (2001) BIND—The biomolecular interaction network database. *Nucleic Acids Res.*, **29**, 242–245.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Benson,D.A., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J., Rapp,B.A. *et al.* (2002) GenBank. *Nucleic Acids Res.*, **30**, 17–20.

Blaschke,C., Andrade,M.A., Ouzounis,C. and Valencia,A. (1999) Automatic extraction of biological information from scientific text: protein–protein interactions. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 60–67.

Chomsky,N. (1956) Three models for the description of language. *IRE Trans. Inform. Theory*, 113–124.

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** (Suppl. 1), S74–S82.

Fukuda,K., Tamura,A., Tsunoda,T. and Takagi,T. (1998) Toward information extraction: identifying protein names from biological papers. *Pac. Symp. Biocomput.*, 707–718.

Hatzivassiloglou,V., Duboue,P.A. and Rzhetsky,A. (2001) Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics*, **17** (Suppl. 1), S97–S106.

Hirschman,L., Park,J.C., Tsujoo,J., Wong,L. and Wu,C.H. (2002) Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, **18**, 1553–1561.

Humphreys,K., Demetriou,G. and Gaizauskas,R. (2000) Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. *Pac. Symp. Biocomput.*, 505–516.

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) The KEGG databases at GenomeNet. *Nucleic Acids Res.*, **30**, 42–46.

Karp,P.D., Riley,M., Saier,M., Paulsen,I.T., Collado-Vides,J. *et al.* (2002) The EcoCyc Database. *Nucleic Acids Res.*, **30**, 56–58.

Krauthammer,M., Rzhetsky,A., Morozov,P. and Friedman,C. (2000) Using BLAST for identifying gene and protein names in journal articles. *Gene*, **259**, 245–252.

Lander,E.S., Linton,L.M. Birren,B., Nusbaum,C., Zody,M.C. *et al.* (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.

Ng,S.K. and Wong,M. (1999) Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Inform. Ser. Workshop Genome Inform.*, **10**, 104–112.

NIH (1999) Relevant terms used for oncogene expression/ pharmacology filters.

Ono,T., Hishigaki,H., Tanigami,A. and Takagi,T. (2001) Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, **17**, 155–161.

Proux,D., Rechenmann,F., Julliard,L., Pillet,V.V. and Jacq,B. (1998) Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. *Genome Inform. Ser. Workshop Genome Inform.*, **9**, 72–80.

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.

Rindflesch,T.C., Hunter,L. and Aronson,A.R. (1999) Mining molecular binding terminology from biomedical text. *Proc. AMIA. Symp.*, 127–131.

Rindflesch,T.C., Tanabe,L., Weinstein,J.N. and Hunter,L. (2000) EDGAR: extraction of drugs, genes and relations from the biomedical literature. *Pac. Symp. Biocomput.*, 517–528.

Schacherer,F., Choi,C., Gotze,U., Krull,M., Pistor,S. *et al.* (2001) The TRANSPATH signal transduction database: a knowledge base on signal transduction networks. *Bioinformatics*, **17**, 1053–1057.

Tanabe,L. and Wilbur,W.J. (2002) Tagging gene and protein names in biomedical text. *Bioinformatics*, **18**, 1124–1132.

Venter,J.C., Adams,M.D., Myers,E.W., Li,P.W., Mural,R.J. *et al.* (2001) The sequence of the human genome. *Science*, **291**, 1304–1351.

Wong,L. (2001) PIES, a protein interaction extraction system. *Pac. Symp. Biocomput.*, 520–531.

Xenarios,I., Salwinski,L., Duan,X.J., Higney,P., Kim,S.M. *et al.* (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids. Res.*, **30**, 303–305.

Yu,H., Hatzivassiloglou,V., Friedman,C., Rzhetsky,A. and Wilbur,W.J. (2002) Automatic extraction of gene and protein synonyms from MEDLINE and journal articles. *Proc. AMIA. Symp.*, 919–923.

Zanzoni,A., Montecchi-Palazzi,L., Quondam,M., Ausiello,G., Helmer-Citterich,M. *et al.* (2002) MINT: a Molecular INTeraction database. *FEBS Lett.*, **513**, 135–140.

Downloaded from https://academic.oup.com/bioinformatics/article/19/16/2046/242199 by guest on 12 March 2021