PROCESSING COMPLEX SENTENCES FOR INFORMATION EXTRACTION

by

Deepthi Chidambaram

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

ARIZONA STATE UNIVERSITY

May 2005

PROCESSING COMPLEX SENTENCES FOR INFORMATION EXTRACTION

by

Deepthi Chidambaram

has been approved

December 2004

APPROVED:

_____, Co-Chair

_____, Co-Chair

_____

Supervisory Committee

ACCEPTED:

_____

Department Chair

_____

Dean, Division of Graduate Studies

ABSTRACT

Genomic research in the last decade has resulted in a huge production of data in the form of microarray experiments, sequence information and publications. The data generated by these experiments are highly connected; the results from sequence analysis and microarrays depend on functional information and signal transduction pathways cited in peer-reviewed publications for evidence. Though scientists in the field are aided by many databases of biochemical interactions available for use online, a majority of these databases are curated by domain experts, often from literature.

Information extraction from text has therefore been pursued actively as an attempt to present knowledge from published material in a computer readable format. An automated extraction tool would not only save time and efforts, but also pave the way to discover hitherto unknown information implicitly conveyed in text. This thesis introduces a way of processing information in biomedical text to result in simple syntactic constructs. The aim is to break up complex sentence structures to processable chunks. Information extraction is made easier by the inherent simplicity and dependencies between the chunks. The module presented is a part of an ongoing work to develop an automatic extraction tool for gene and protein interactions in biomedical text. The experimental results show that the proposed approach significantly improves the recall of the extraction system.

To my parents

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Hasan Davulcu and Dr. Chitta Baral, who have encouraged and guided me throughout the process, and nudged me toward the right path whenever I wandered for too long. I am also grateful to Dr. Yoganand Balagurunathan for encouraging and supporting me in my work. My sincere thanks to Dr. Edward Suh of Translational Genomics Research Institute (TGen) and the scientists at TGen for providing the guidance and feedback on my work. I would also like to thank Dr. Andrew Rzhetsky and Dr. David Corney for providing us with the data necessary to perform an accurate evaluation of our system. I would also like to thank Syed Toufeeq Ahmed and Ravi Bhimavarapu, my co-workers in this project for sharing their code and comments. Lastly, I am highly indebted to my family and friends for their support and encouragement.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

Introduction

Genomic research is the last decade has resulted in the production of a large amount of data in the form of microarray experiments, sequence information and publications discussing the developments. The data generated by these experiments are highly connected; the results from sequence analysis and microarrays depend on functional information and signal transduction pathways cited in peer-reviewed publications for evidence. Though scientists in the field are aided by many databases of biochemical interactions available for use online, a majority of these databases are curated by domain experts, often from literature. Information extraction from text has therefore been pursued actively as an attempt to present knowledge from published material in a computer readable format. An automated extraction tool would not only save time and efforts, but also pave way to discover hitherto unknown information implicitly conveyed in text. Once presented in a processing-friendly format, the consolidated knowledge can be used to reason molecular functions for unknown genes, and identify newer and better drugs for treatment. Current extraction tools employ diverse approaches in aim for this milestone.

Extraction systems in the biomedical domain are designed for different textual contents such as the patient trials, clinical records and pathological reports in addition to publications. Information extraction systems for literature work on unstructured text as

opposed to the semi-structured nature of the reports and trial studies. Work in this area has focused on extracting a wide range of information such as chromosomal location of genes, protein functional information, associating genes by functional relevance and relationships between entities of interest. While clinical records provide a semi-structured, technically rich data source for mining information, the publications, in their unstructured format pose a greater challenge, addressed by many approaches. With the contributions to Medline growing at the rate of 1500 abstracts per day [1], the need for an automated system becomes inevitable. The thesis presented is a part of an ongoing work to develop an automated information extraction system for biomedical literature.

The corpus used for extraction in this work is the PubMed[1] database from National Library of Medicine (NLM). The PubMed database contains 11 million citations to articles from biomedical journals. The abstracts of the articles are freely available and are chosen as the corpus of this work due to their concise representation of the information in the articles. The sentences in the abstracts are written in creative ways, ranging from the simple to compound-complex. Hence, an extraction system designed on these abstracts should take advantage of the sentence structures of the abstracts to isolate the lines of thoughts conveyed in the sentence. This thesis focuses on processing complex sentence structures in text to extract interactions between genes and proteins using the Link Grammar Parser.

The thesis documents our work on two approaches to processing complex sentences to extract gene interactions from text - a rule building approach where a domain expert writes the extraction rules for the sentence patterns, and an automated approach using the dependencies between the words in a sentence. The manual system involves learning patterns of extraction for simple sentences which are used by a complex sentence processor

[1] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

written by the author. The learning system for the simple sentences was written by Syed Ahmed. The automated system was a group effort, with contribution by Ravi Bhimavarapu in pronoun resolution and the Complex Sentence Processor written by the author. The complex sentence processor makes use of the Link Grammar Parser of Sleator and Temperly from Carnegie Mellon university. Syed Ahmed developed a system to process the simple sentences output by the complex sentence processor to extract interactions.

CHAPTER 2

Background and Related Work

Information extraction from text has been pursued by various researchers over the years. Peer reviewed publications provide details on the problems and approaches in the area. This chapter provides a summary of the problems and challenges faced by information extraction systems in the field. The success of information extraction depends on the performance of the various subtasks involved. These include information retrieval, text categorization, entity identification and relation extraction. While current systems aim at analyzing relationships in literature, a possible future direction is the detection of signal transduction pathways, and perhaps prediction and reasoning on the pathways extracted. This chapter provides a discussion of the trends in each of the sub-tasks. Figure 1 gives an overview of the subtasks in information extraction [1].

1. Text Retrieval and Categorization

Information retrieval is the identification of documents of interest. Research in information retrieval includes searching, indexing, categorization or clustering and visualization of documents. The simplest querying systems provide a boolean query as input to retrieve

[1] The ISMB 2004 tutorial on information extraction from biomedical text provides a good survey of the process.

documents of interest. The keywords with the boolean operators applied on them are used to identify the documents that satisfy the search constraints. The boolean query methodology is followed by literature search interfaces like PubMed. The keyword retrieval, while simple, returns a large number of documents. Since the retrieval is purely based on keyword matching, there is a high change of irrelevant document returned in the search results, as well as relevant documents being missed out due to various factors such as spelling variation and synonymy. Vector space models serve to alleviate this problem by posing a similarity measure between the documents and keywords in the query. The similarity queries of the vector model treat both the results and the query terms as individual documents and computes the relevance of the result based on the similarity measure between the documents. Document cosine similarities and TF-IDF factors are the commonly used similarity measures in this approach [2]. Vector space models tend to provide less irrelevant results than plain keyword matching method. Probabilistic models provide a 'soft' retrieval method by returning documents that satisfy the query with high probability rather than the exact matches preferred by the previous approaches. While these approaches provide increasingly efficient ways to retrieve documents, they do not fare well in cases of synonymy (different words of the same meaning) and polysemy (multiple context for the same word). Latent Semantic Indexing(LSI) introduced by Dumais et al [3, 4] achieves to take care of the problem of synonyms and multiple meanings by the use of singular value decomposition. In LSI, the document vector is reduced to the most significant singular values which are then used to represent the hidden semantics of the documents and queries. A drawback in the LSI approach is the loss of intuitive association between the query words and the documents. The user is no longer able to identify the query terms corresponding to the documents as the results are not expected to contain the exact words anymore.

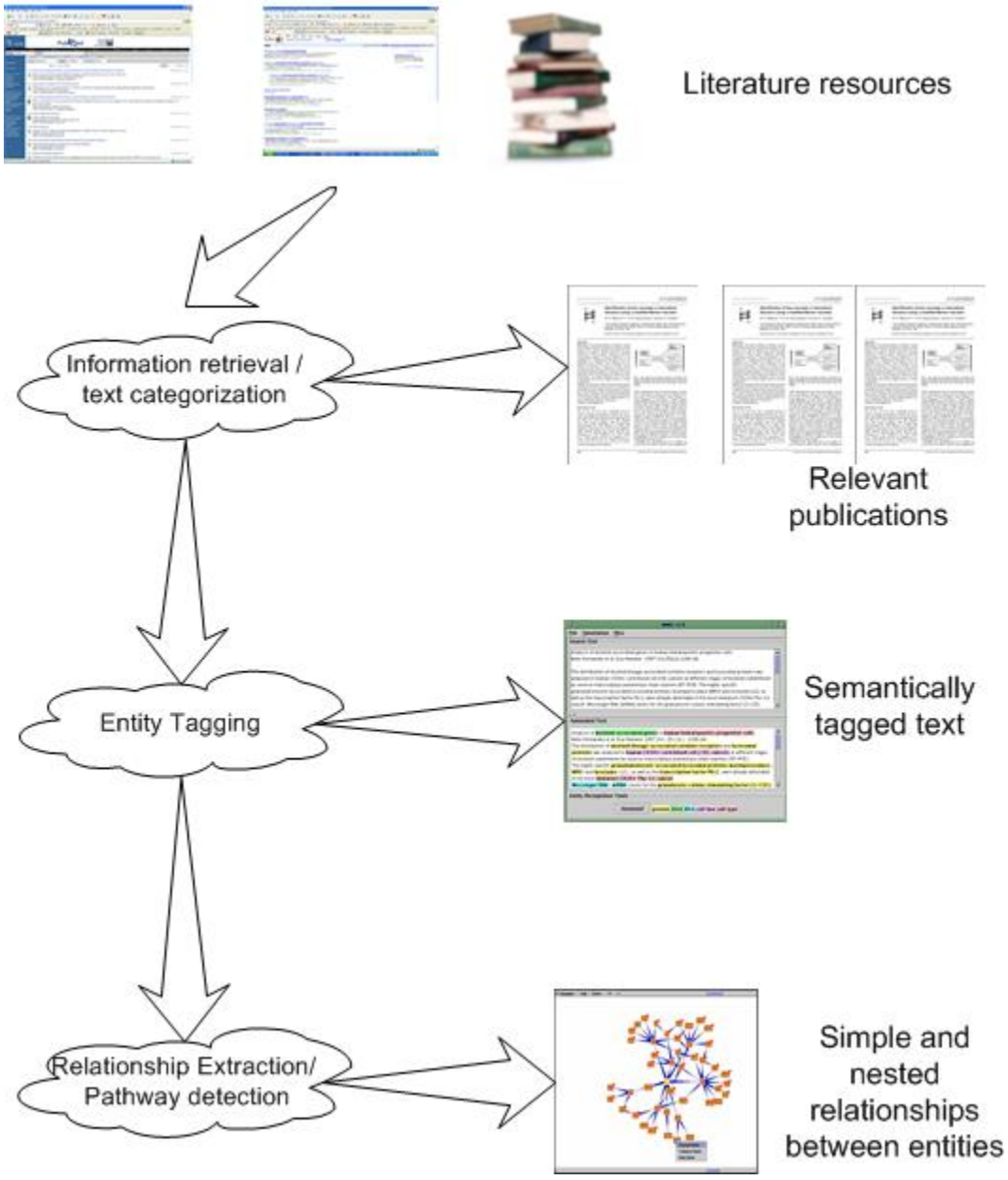

Figure 1. Information Extraction: subtasks

Text Categorization has seen the application of both Knowledge Engineering and Machine Learning approaches in classifying documents based on the information contained. While Knowledge Engineering approaches rely on a domain expert to specify the classification rules, machine learning approaches are automated and prevalently use various clustering and classification algorithms. Conventional classification algorithms are often augmented using feature selection procedures that enhance the categorization of documents. Word co-occurrences with ontologies used for the semantics of the words are also used in the classification of documents. The CONSTRUE [5] system follows the knowledge engineering approach, where the rules are specified as a disjunction of conjunctive clauses.

2. Named Entity Recognition

Entity extraction or Named entity identification is the process of identifying the words or phrases of interest such as genes, proteins, protein families, drugs, chemicals and pathways in text. Entity identification has also been thoroughly researched over the years, with various challenges such as the BioCreative[2] and shared tasks in conferences[3] addressing the issues and evaluating the performances using a common corpora. The simplest and frequently used approach to entity identification is a dictionary matching approach where the entity names are compiled as a dictionary and a string match with an entry in the dictionary tags the words or phrases as gene or protein names. A variety of publicly available databases provide the resources for entity names. NCBI's LocusLink[4], HUGO[5], SwissProt[6] are among the databases that provide gene and protein names and their synonyms.

[2]http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html
[3]http://research.nii.ac.jp/ collier/workshops/JNLPBA04st.htm
[4]http://www.ncbi.nlm.nih.gov/projects/LocusLink/
[5]http://www.gene.ucl.ac.uk/nomenclature/
[6]http://www.ebi.ac.uk/swissprot/

Entity Identification systems generally use rule based approaches and machine learning techniques to mark the phrases of interest in text. Rule based approaches rely on regular expressions and heuristic rules to identify gene names. Fukuda et al [6] follow a combination of regular expressions and expansion rules to identify single word and multi-word gene names. [7] also follow a rule based approach to identify biological entities in text. Some of the machines learning approaches followed for NER include decision trees, Bayesian classifiers, hidden markov models, iterative error reduction, boosted wrapper induction and support vector machines. The ABGene system from Tanabe and Wilbur [8] uses the Brill's tagger to learn transformation rules to tag the gene and protein names in text. The rules are based on the word occurrences, neighboring words and part of speech tags of the words and the neighbors.

Research in entity recognition has resulted in the development of various corpora for the purpose of providing a benchmark for the entity recognition systems. The GENIA corpus, a hand -annotated corpus of abstracts from over 2000 medline articles on human blood transcription factors uses the GENIA ontology to tag concepts in text. The recent JBLPBA challenge used the GENIA corpus as the test data for its shared task on Entity recognition. The participants in the task used various machine learning approaches, sometimes using a combination of approaches such as the support vector machines and hidden markov models of Zhou [9]. The results from the task can be obtained from their webpage[7]. The BioCreative[8] corpus is more general in nature, deliberately constructed with challenging false positives by the National Library of Medicine.

[7] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html
[8] http://www.mitre.org/public/biocreative/

3. Identification of Relationships in Text

Information extraction is the process of locating assertions between entities of interest in text. The current focus of extraction systems is to identify protein interactions from text. Blaschke [10] and Ono [11] are some of the systems that use rule based approaches to extract protein interactions from text. Ono et al [11] use regular expressions with protein name dictionaries to extract protein-protein interactions. The text in their approach are processed through a protein name identifier and then through a module which processes complex and compound sentences. Regular expressions are used to identify the structure of complex and compound sentences with the part of speech tagging done using the Brill's [12] part of speech tagger. The SUISEKI system of Blaschke [13] uses predefined frames on the sentences to extract interactions. The frames are assigned a probability score based on their reliability. MedScan, a natural language processing algorithm [14] is used in the PathwayAssist system to identify the interactions between the genes and proteins. The system uses protein name dictionaries to pre process the text. The interactions are identified by a Unification grammar that uses transition networks. GENIES [15], a partially implemented natural language extraction engine, also uses a term tagger and pattern matcher. The grammar used to match the patterns is based on syntactic and semantic constructs. The GENIES system has been extended as GeneWays [16], an information extraction system with a web interface that allows the users to search and submit papers of interest for analysis.

The BioRAT system [17] uses manually defined templates that combine lexical and semantic information of the words to identify protein interactions. The system also provides means for the user to specify additional templates. Several Gazetteers of entity names

derived from biological sources are used to mark up entities. The BioRAT system uses the GATE [18] natural language processing system from the Natural Language Processing Research Group at the University of Sheffield. The GENIA corpus has also been used to analyze the approach of using manually created context free grammars and probabilistic context free grammars for information extraction in [19]. The authors have created a decision tree type dependency parser to analyze text for protein interactions. While the parser rules are very similar to this thesis' approach of analyzing the linkages, they are restricted by the non-scalability issue accompanying manual approaches. Furthermore, they do not address complex sentence structures, which are taken advantage of in this thesis.

Despite the fact that using part of speech structure is an intuitive approach to form extraction rules, other properties of sentence structures can also be used. [20] Addresses the problem of extracting protein interactions by filtering the input text into a stream of tokens and using an extendable but manually built Context Free Grammar (CFG) that is designed specifically for parsing biological text. The CFG is based on prepositions, protein names and transitive words. [21] extract interactions from biomedical text using preposition-based templates. Partial grammars based on prepositional phrase attachments are used to identify the frequent patterns and extract information based on the attachment of the phrases. They address the issue of co-reference by filling up references based on the templates of the prepositions that were merged to create interactions. Chiang et al [22] propose the GIS system which extracts relations between genes using a decision tree approach. The sentence patterns in terms of wording and term distribution in describing relations are represented as a variant of decision tree. The system extracts relations classified into three categories - positive, cooperative and negative.

Manual rule engineering approaches in information extraction can be labor-intensive

and skill-dependent. [23] explored an automatic rule-learning approach that uses a combination of FOIL [24] and Nave Bayes Classifier to generate rules. They make use of the linguistic structure of the text, and a Nave Bayes Classifier to characterize the words in sentences and phrases. To incorporate Nave Bayes to FOIL, they have, for instance a Nave Bayes predicate for protein, so that the extracted sentence must contain words that are classified as protein. These extra Nave Bayes predicates are given to FOIL as candidates of literals for forming the extraction rules. [25] use FOIL to learn rules to extract information in the context of global warming domain. Rather than using the linguistic structure of sentences, it uses an ontology-based approach. Words such as "Carbon Dioxide" would be in part of the ontology. With the ontology, FOIL is used to form rules based on co-occurrences of words. While the automatic rule learning approaches mentioned use an existing rule-learning system that mainly based on linguistic structure, [26] use a grammatical approach to develop their own rule-learning system to learn extraction rules automatically. Rather than examining the constituents or categories a word belongs to, it uses link grammar, which is based on a model that words within a text form links with one another. The property of link grammar allows their system to avoid dealing with complex and co reference resolutions.

A survey of techniques used in protein name extraction and protein interaction extraction is presented in [27]. Their protein interaction extraction methods include extensions of the techniques used for name extraction, and heuristic rules to identify the interactions. Their techniques rely on word occurrences, while ours are based on links between words. Extraction systems have also used link grammar to identify interactions between proteins [28, 29]. Both these approaches identify gene and protein names and define an interaction as any arbitrary link that connects these two words. The linguistic

roles of the words in the sentence are not taken into account.

While most of the IE systems mentioned above focused on extracting interactions between genes, biologists are also interested in their corresponding pathways. Besides, extracting interactions between genes alone without information such as locations on where the interactions occur can be misleading to biologists. We utilize various biomedical ontologies such as, Gene Ontology, UMLS and WordNet [30] to overcome the burden of labor intensive pattern engineering and to enable automated extraction of complex events.

4. Sentence Structures and the English Grammar

Analysis of sentence structures provides a means of identifying the semantics of the sentence through the roles played by the syntactic components. The approaches followed by the rule engineering and the automated systems to process complex sentences are based on the sentence structures defined in the English language. This section provides a brief overview of the English grammar and sentence structures.

Sentences in the English language are defined as following the 'SVO' construct, where each sentence has a subject and a predicate. A clause is defined as a sentence with at least a subject and a verb. The sentences in English are classified as either simple, complex, compound or complex-compound based on the number and types of clauses present in them.

A simple sentence is a single independent clause which conveys a single thought. The simple sentence has one lexical verb, and one or more auxiliaries. For instance, the sentence below conveys a single idea and has one lexical verb - ringing.

```
A telephone was ringing in the darkness.
```

```
Phosphorylation by ATM activates c-Abl.
```

A compound sentence, on the other hand contains many lexical verbs. Formally, compound sentences are independent clauses connected by a co-coordinating conjunction, semi-colon or an independent marker such as 'however', 'therefore' and 'moreover'. Each of the clauses in a compound sentence can function as simple sentences by themselves and do not rely on each other to convey the meaning intended.

```
The hostess began reading choice excerpts from the inane article and
Langdon felt himself sinking lower and lower in his chair.
```

```
c-Abl phosphorylates tyrosines in the C-terminal domain (CTD) of
RNA polymerase II (RPase II; Km 5 0.5 mM); the c-Abl SH2 domain
is a specificity determinant for this reaction.
```

The clauses in all sentence types may also have a single word or multiple words as subject, verb or object; Multi word components are generally referred to as compound subjects, compound verbs and compound objects. A complex sentence has one independent clause and one or more dependent clauses that rely on the some component of the independent clause for their completeness. The clauses may be connected by dependent markers such as 'though', 'although', 'if', 'when' etc.

```
c-Abl tyrosine kinase activity is blocked by pRb,
which binds to the c-Abl kinase domain.
```

Complex-compound sentences as the name suggests, are a combination of the complex and the compound sentence types. The complex-compound sentences have one or more independent clauses and one or more dependent clauses.

```
In addition to a constitutively occupied E2F1-Sp1 site immediately
```

```
upstream of the cyclin E transcription start region, there is downstream
a cell cycle-regulated site (termed CERM) that may function as a
cyclin E-repressor module.
```

As seen from the examples, sentences from biomedical abstracts are seldom simple in nature. By the nature of the sentences, multiple ideas are conveyed in a single sentence. Complex, compound and complex-compound sentences are collectively referred to as 'complex' sentences in this work. Processing complex sentences and converting them into independent clauses ensure the separation of the thoughts conveyed in the sentence. These clauses can then be processed as simple sentences to extract the information they offer. Two approaches to complex sentence processing are presented in this thesis - a manual rule building approach and an automated approach using Link Grammar.

CHAPTER 3

Manual Rule Building Approach in Complex Sentence Processing

Our initial work on complex sentence processing was a manual rule engineering approach. The sentence structures of complex sentences were analyzed and processing rules were written in prolog to extract interactions from text. The rule engineering approach was later dropped in favor of the automated processing approach for lack of scalability and reliance on the skills of the domain expert. This section gives an overview of the rule engineering approach.

1. The system architecture

The rule engineering approach was largely based on part of speech tags of words. The goal was to extract interaction predicates in the form of 3 arguments from curated text. An overview of the system is given in Figure 2. The corpus for the manual approach was text describing the molecular pathways of mammalian cell cycle and control from an article in Molecular Biology of the Cell by Kurt Kohn [31]. An off-the-shelf natural language processor from Infogistics[1] was used to process the curated text and generate the part

[1]The Infogictics NLProcessor, http://www.infogistics.com/textanalysis.html

of speech tags. The NLProcessor from Infogistics is the windows version of the natural language processor of the Language Technology group based on the Penn Treebank tagset.

The system works on an iterative process of improving the extraction efficiency by intervention of a rule engineer. The initial training data of interactions were marked by the biologist using a user interface designed in Java, the Example Marker system. The rules were built using the knowledge from the training data and dictionaries serving as the knowledge bases.

Dictionaries of protein and gene names and interaction words were used to restrict the 'interactions' extracted from the text. The protein name dictionary was compiled from LocusLink, the NCBI database for gene information. The names and their aliases were extracted from the database using a perl script. The extracted list of about 17,000 names was then processed to form a prolog knowledge base.

Analysis of our data showed that the majority of the interaction words were verbs. So, the words in UMLS were matched against the verbs in WordNet lexical dictionary to come up with a list of biological verbs. The list was then parsed through a stemming algorithm (the Porter Stemmer algorithm [32]) to account for all word forms. Additions to this list were done manually if interaction words were found to be missing.

2. Rules for extracting interactions based on sentence structures

Extracting information from simple sentences is a trivial process. The Example marker system learns part of speech patterns corresponding to the interactions that occur in the sentences. The patterns form a shallow parse which addresses sentence patterns in simple sentences. Complex sentence patterns require more processing, which is addressed by hand built rules.

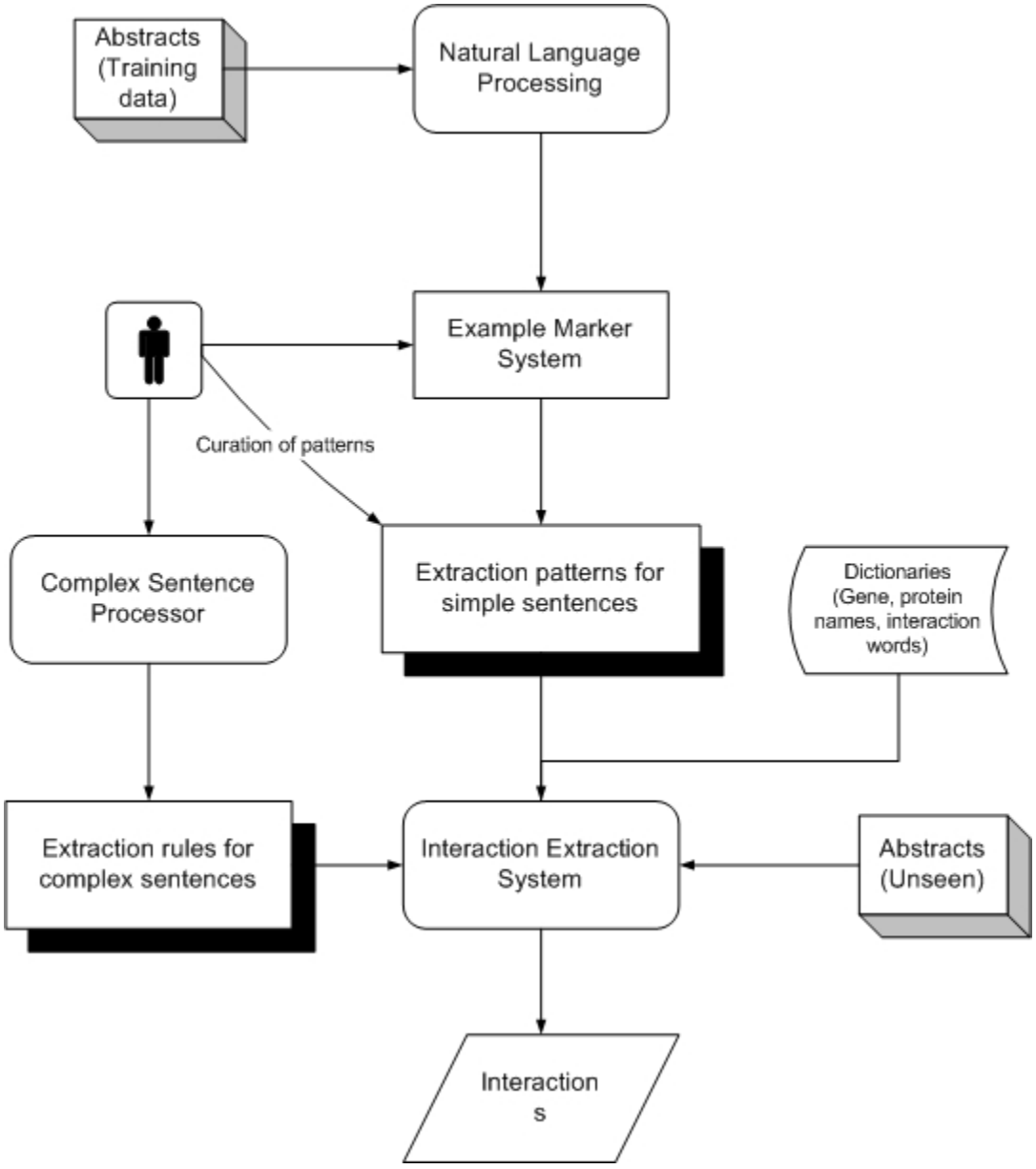

Figure 2. System Overview: Rule Engineering Approach

The system built is a prolog system which is accessed through a java interface which combines the various modules. The part of speech patterns for the interactions are learnt using the Example Marking system. Sample patterns learnt by the user interface are given below. The Example Marker GUI was written by Toufeeq to learn the patterns from simple sentences. The code for processing rules for based on these patterns for simple sentences was also contributed by Toufeeq.

The patterns specify the part of speech tags possible for each of the three arguments as a list and the format of the interaction to be extracted (the second argument in the extract predicate). The sentences were then processed to replace pronouns and co-references. The pronoun resolution was done manually to replace the references by the nouns / noun groups being referred to. The rules were written based on sentences structures such as simple sentences connected by conjunctions, punctuations etc. The complex sentences were split and rewritten as simple sentences by the application of these rules. The simple sentences were processed on the patterns to get the three argument interactions.

```
extract([word([tag = NNP],_h13160),word([tag = VBZ],_h13161),
word([tag=JJ],_h13162)],
interact(_h13160,_h13161,_h13162),true).
extract([ng(_h99513),vg(_h99514),ng(_h99515)],
interact(_h99513,_h99514,_h99515),true).
extract([ng(_h108321),vg(_h108322),word([tag=NNP],_h108323)],
interact(_h108321,_h108322,_h108323),true).
extract([ng(_h158006),vg(_h158007),word([tag=RB],_h158008)],
interact(_h158006,_h158007,_h158008),true).
extract([ng(_h241968),vg(_h241969),word([tag=VBG],_h241970)],
interact(_h241968,_h241969,_h241970),true).
```

2.1. Sentence structures addressed by the rules.

1. Resolution of 'and' - multiple rules:

   Simple sentences connected by connectives 'and', 'but', 'yet' and 'or' are split at the connective and processed as separate sentences.

```
Sample Input:
  The Cyclin D1 promoter is activated by E2F4,
  but the cyclin D1 promoter is repressed by E2F1 via pRb

Sample output:
  interact([The,Cyclin,D1,promoter],[is,activated],E2F4).
  interact([the,cyclin,D1,promoter],[is,repressed],E2F1).
```

2. Resolution of 'and' - in arguments (multiple rules). The connectives 'and', 'or' may occur in the place of arguments too, as in the case of

   ```
   Cdk4 and Cdk6 bind exclusively to D-type cyclins.
   ```

   The solution is to split the sentence at the connective.

   ```
   Cdk4 bind exclusively to D-type cyclins.
   Cdk6 bind exclusively to D-type cyclins.
   ```

   A list processing operation will do this. The output from this processing is,

   ```
   interact(Cdk4,[bind,exclusively],[D-type,cyclins]).
   interact(Cdk6,[bind,exclusively],[D-type,cyclins]).
   ```

3. Occurrence of 'thereby' (multiple rules) Two simple sentences, when separated by a 'thereby' clause have a causal relationship between themselves. They are separated at thereby and the causal relationship is written out once their processing is done.

   ```
   Sample input:
   CycH:Cdk7 (also known as Cdk-activating kinase [CAK]) phosphorylates
   a site on the T-loop of Cdks and thereby CycH:Cdk7 causes the loop
   to be displaced to allow access to the catalytic site.


   Sample output:
   interact(CycH,phosphorylates,[a,site])
   causes
   interact(CycH,causes,[the,loop]).
   ```

4. Rule for reactions separated by 'and'. Normally, verb groups specifying reactions separated by a connective would be a single verb group. But our NLP tags them separately, so an explicit rule has to be written to handle this case.

```
Sample input:
c-Abl binds and tyrosine phosphorylates paxillin
in an adhesion-dependent manner.

Sample output:
interact(c-Abl,binds,paxillin).
  interact(c-Abl,[tyrosine,phosphorylates],paxillin).
```

5. Simple sentences with a semicolon. The sentence after the semicolon is considered as extra information for the sentence preceding it.

```
Sample input:
Cdc25A may be transcriptionally activated by c-Myc;
the Myc:Max heterodimer binds to elements in the Cdc25A gene.

Sample output:
  interact(Cdc25A,[may,be,transcriptionally,activated],c-Myc)
  moreover
  interact([the,Myc],binds,[the,Cdc25A,gene]).
```

6. Pronoun resolution in 'which'. The subject of the sentence after 'which' is the word or group preceding it.

```
Sample input:
  Another DMP1-regulated gene is CD13/aminopeptidase N,
  which is activated cooperatively by DMP1 and c-Myb;
  CD13/aminopeptidase N activation by DMP1 is inhibited
  by cyclin D independent of Cdk4/6.

Sample output:
  interact([CD13/aminopeptidase,N],[is,activated,cooperatively],DMP1)
  moreover
  interact(CD13/aminopeptidase,[is,inhibited],[cyclinD,independent]).
```

7. List of arguments - a comma separated list or a list separated by conjunctions. The sentence is rewritten with each of the arguments. Each rewrite is processed as a single sentence.

```
Sample Input:

  Dyhydrofolate reductase (DHFR) is activated via the E2F
  transactivation domain, whereas B-myb, Cyclin E, E2F-1,
  E2F-2, and Cdc2 are regulated via the repression domain
  of pRb family proteins.

Sample output:
  interact([Dyhydrofolate,reductase],[is,activated],
     [the,E2F,transactivation,domain]).
```

```
interact(B-myb,[are,regulated],[the,repression,domain]).

interact([Cyclin,E],[are,regulated],[the,repression,domain]).

interact(E2F-1,[are,regulated],[the,repression,domain]).

interact(E2F-2,[are,regulated],[the,repression,domain]).

interact(Cdc2,[are,regulated],[the,repression,domain]).
```

8. Other rules that haven't been used in the examples processed so far.

   (a) Simple sentences separated by connecting words that specify causal relationships

      i. In response to, stimulates, requires, induces etc (positive relation)

      ii. Blocks, inhibits etc (Negative relation)

2.2. Sample rule for a list of agents.

```
extractListEnd(_,[]).

extractListEnd(S1,[X|Y]) :-

conc(S1,[X],S),extractComplex(S),

extractListEnd(S1,Y).

extractListStart([],_).

extractListStart([X|Y],Part) :-

conc([X],Part,Whole),extractComplex(Whole),

extractListStart(Y,Part).

extractComplex(S):-

contains([W1,word([_],',',),_,word([_],',',),_],S),

splitList([W1],S,S1,S2),

S1=[],getList(S2,List),

length(List,Len),ith(Len,List,Ele),

sublist([Ele],S,Part1,Part2),

extractListStart(List,Part2),!.

extractComplex(S):-

contains([W1,word([_],',',),_,word([_],',',),_],S),

splitList([W1],S,S1,S2),

S1\=[],getList(S2,List),

extractListEnd(S1,List),!.
```

The gene and protein names list was reduced by a regular expression matching module which reduced the list of 112810 names and aliases to 311 names and aliases that

were contained in the text we used. A few facts from the dictionary are given below.

```
isa('3''-nucleotidase',gene).

isa('3'' repair exonuclease 1',protein).

isa('E2F4',gene).

isa('5''-nucleotidase, cytosolic IA',protein).

isa('M1',gene).

isa('PP2A, subunit B''',protein)

interaction('accumulat').

interaction('activat').

interaction('elevat').

interaction('hasten').

interaction('incite').

interaction('increas').

interaction('induc').
```

The extracted three argument interactions were restricted to contain words in the interactions list in the second argument of the interact predicate. The first and the second arguments or a part of them were conditioned to appear in the gene and protein names list or the interactions list, to allow arguments like 'the c-abl kinase activity' and 'phosphorylation' to occur in addition to gene and protein names.

The system was tested on curated text from a different part of the Kohn article [31]. The texts from the article were rewritten after pronoun resolution. The testing data is then processed by the natural language processor and the processor outputs the tagged text with the part of speech tags in xml format. The XML representation is transformed to the

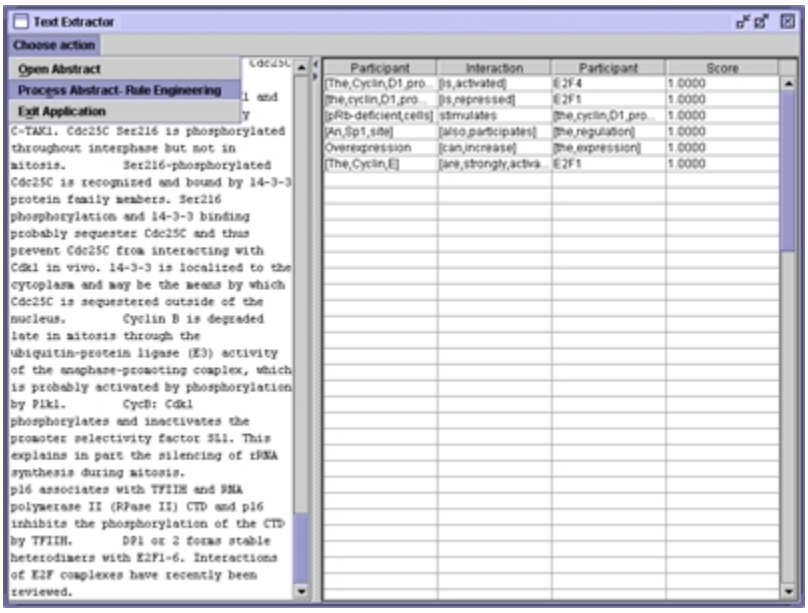

Figure 3. Snapshot of the Rule Engineering system

list representation of Prolog using a DOM parser, which parse the XML file and creates a DOM tree. Tree traversal and manipulations yield the list format. The list contains the tag and the word for the entire sentence. This list representation is passed on to the complex sentence resolving module. Figure 3 gives a snapshot of the system in use.

3. Results of the rule engineering approach

Figure 4 shows the output of the extraction system for the sample input. The sentences in the text were broken down to simple sentences and matched against the patterns learned by the example marker. The interactions were extracted based on the patterns. The Kohn paper has explanations to a pathway diagram divided into many sections. The rules and the patterns were written incrementally on these sections. In the first iteration, Parts A and B were taken as the training set and the rules and the patterns were written for the sentences in these parts. This was then tested against C. The rules had a precision

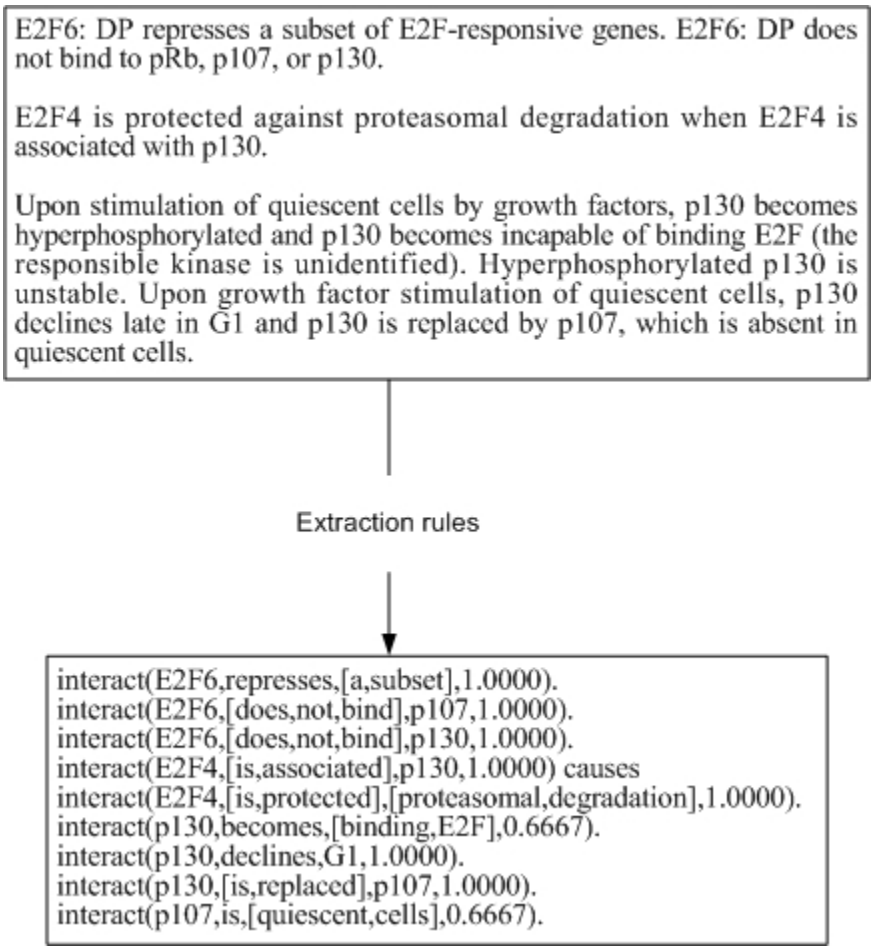

Figure 4. Sample Run of the Rule Engineering system

and recall of 57% and 43% respectively. The text in part C was then used in training the complex sentence resolution module. Rules were written for the complex sentence patterns in the text. Patterns of part of speech were also added pertaining to the text in part C. The new set of rules and patterns were tested against the text in part E. The training data for the combined sections of A, B and C consisted of 62 curated text. This was tested against 30 curated texts from part E. The precision for the data was 82.3% and the recall was 71%.

4. Discussion

As seen above, the rule engineering approach involved a lot of effort and time. In addition, the approach is solely dependent on the skills of the person encoding the rules. An extension of the system to a different corpus or domain would require extensive re-writing of the rules. Hence, because of these issues on scalability and efficiency, an automated processing system for complex sentences was sought.

CHAPTER 4

The System Architecture

The previous chapter discussed a rule engineering approach to extract information from complex sentences. While the manual approach ensures precision of the patterns due to the intervention of the domain expert, it is not scalable and is highly dependent on the skills of the knowledge engineer. Hence an automated approach to complex sentence processing is pursued.

The architecture of the automated system for complex sentence processing is shown in Figure 6. The complex sentence processing system takes the abstracts as input and gives a tab separated database of simple sentences as the output. The sub-systems that are involved in the process are detailed in the figure. The abstracts from PubMed are processed to replace pronouns. The resolved abstracts are then sent to the Entity Tagger which tags the words and phrases in the abstracts as gene, proteins, chemicals or locations using the dictionaries derived from various ontologies. The tagged sentences are then run through a pre-processor module that corrects the sentences to improve the parsing of the sentences by the link grammar parser. The link grammar parser processes the filtered sentences and provides the linkages for the sentences to the Complex Sentence Processor module. The complex sentence processor module analyzes the linkages and uses them to convert the sentences in the abstracts to the internal clause format representation, breaking down

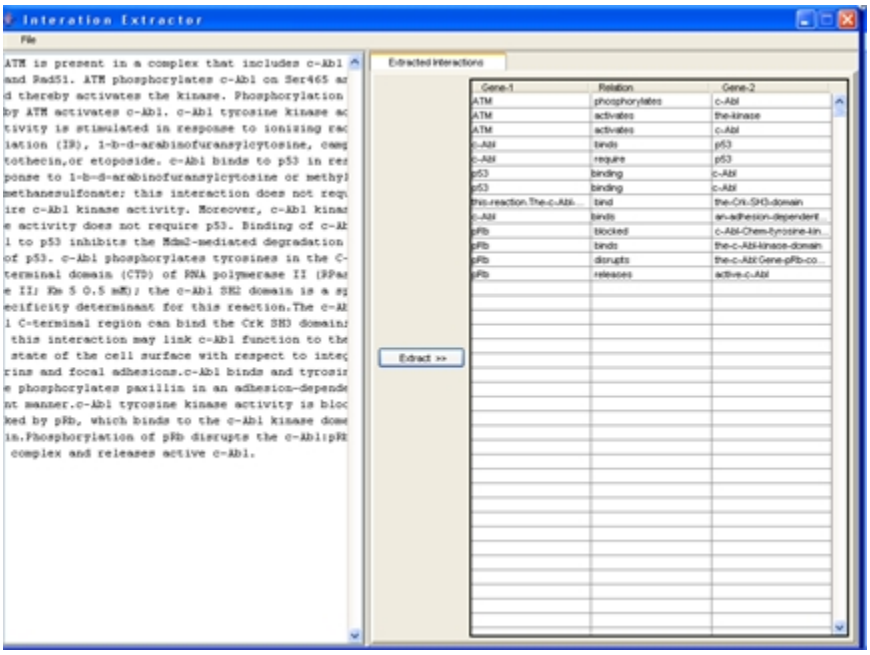

Figure 5. Snapshot of the Automated Interaction Extraction System

complex sentences. The linkages from the link grammar are obtained using a link grammar wrapper developed by Syed Toufeeq Ahmed.

The simple sentences are maintained in the clause format as bar separated files. These sentences are then processed by an Interaction Extractor module that uses linguistically significant combinations of the components of the clause to extract the interactions between genes and proteins mentioned in the text. The interaction extractor was developed by Toufeeq. Chapters 6 and 8 focus on the sub-systems of the Complex Sentence Processor. A snapshot of the current version of the GUI for the automated information extractor is given in Figure 5. The flow of code and data is given in Figure 25 of Appendix B.

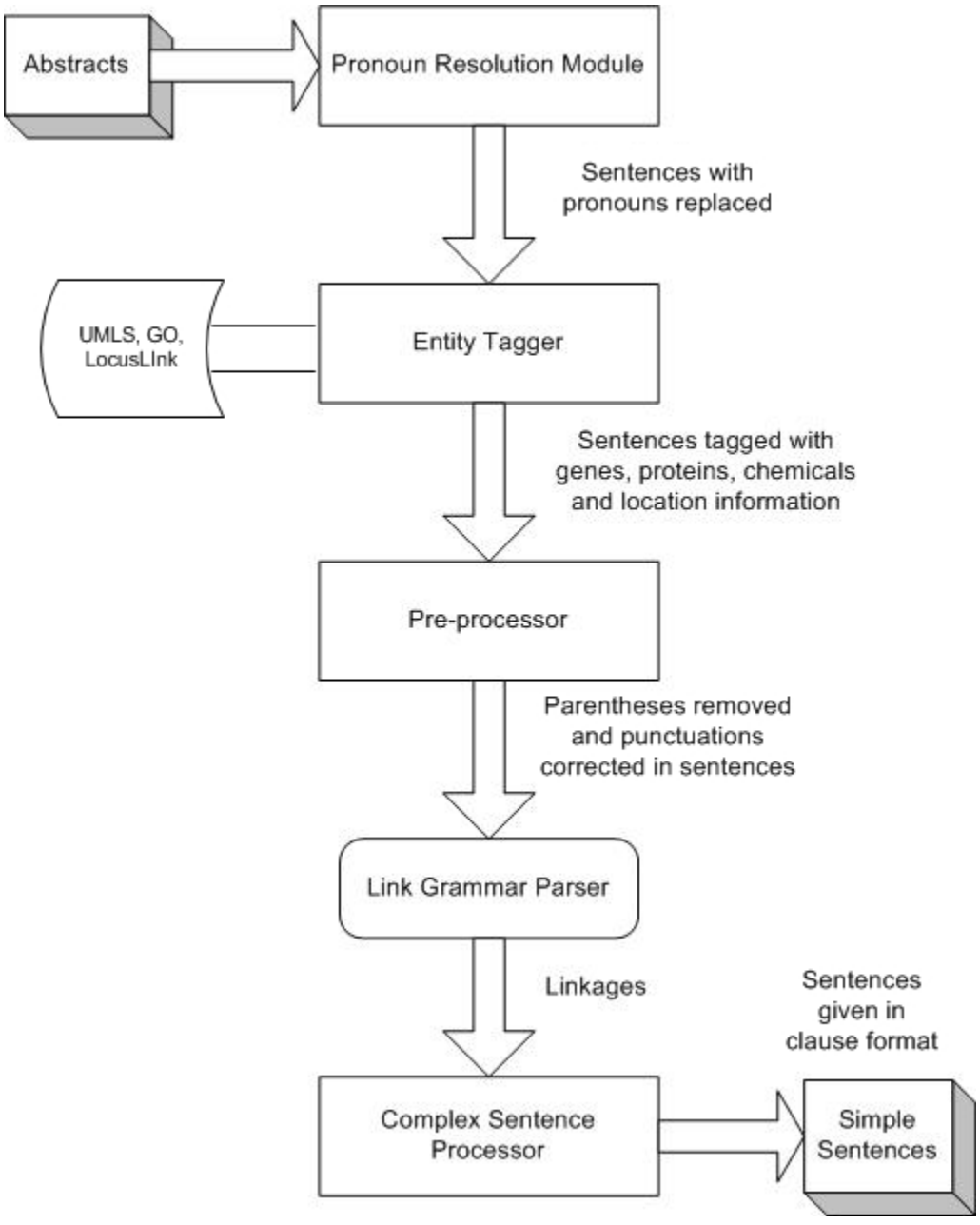

Figure 6. System Overview: Automated Approach

CHAPTER 5

Processing Text for Complex Sentence Processor

The Complex Sentence Processing module takes text from the abstracts as the input and outputs a simple sentence database for each abstract, indexed by the abstract identifier and the sentence identifier. The text before being passed to the CSP algorithm, however, has to be pre-processed. The pre-processing step includes pronoun resolution, entity tagging and pre-processing the tagged sentences to improve parsing results in the Link Grammar Parser. This chapter details each of these processes in detail.

1. Pronoun Resolution

Interactions are often specified through pronominal references to entities in the discourse, or through co references where a number of phrases are used to refer to the same entity. Hence, a complete approach to extracting information from text should also take into account the resolution of these references. References to entities are generally categorized as co-references or anaphora. Co-reference resolution involves splitting the noun phrases in a text to clusters of *equivalence classes*. An illustration is given in Figure 7.

Anaphora resolution, on the other hand deals with identification of pairs of referent noun phrases in the text. Pronominal anaphors deal with identifying the noun phrases

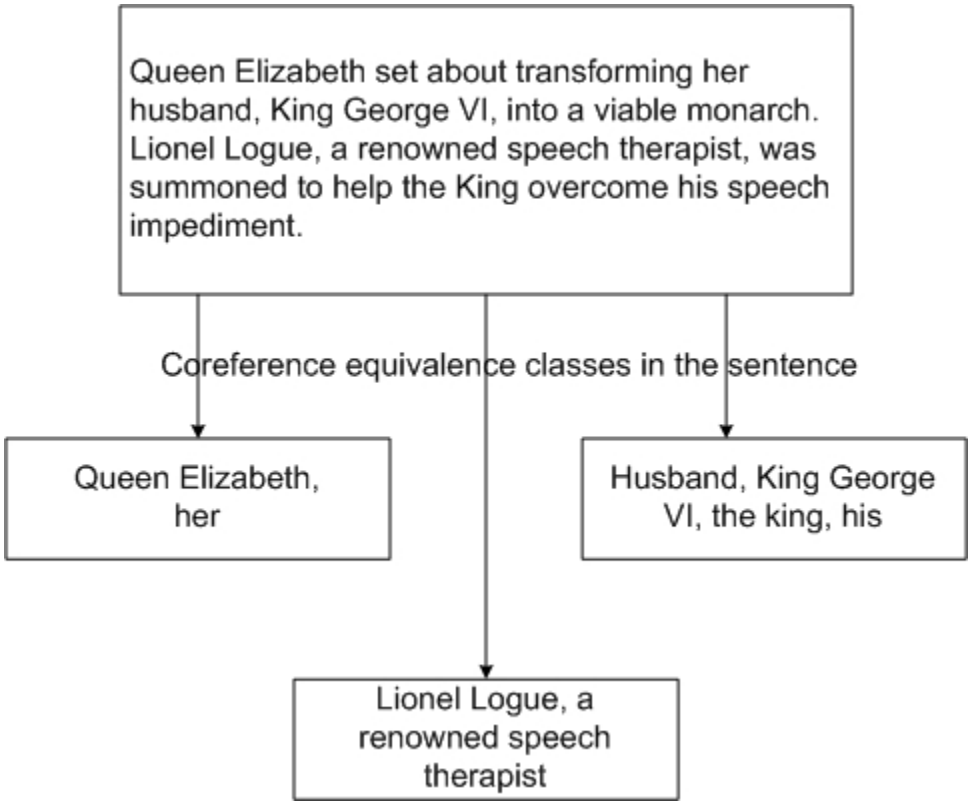

Figure 7. Coreference Equivalence Classes

pointed to by the pronouns. Noun phrases that refer to other noun phrases are classified as sortal anaphora. Event anaphora refers to an event previously specified in the text. Figure 8 illustrates each of these types of anaphora. While co-reference resolution is a phenomenal problem, anaphora resolution is relatively simpler to solve and has been addressed by various approaches [33]. The anaphora resolution system currently focuses on pronouns. Furthermore, we have observed that entities in literature are referred to only by third person pronouns, and the first and second person pronouns are used to refer to the authors of the papers. Hence the current system focuses on third person pronouns and reflexives.

The pronoun resolution system uses a heuristic approach to identify the noun phrases referred by the pronouns in a sentence based. The heuristic is based on the number of the pronoun (singular or plural) and the proximity of the noun phrase. The first noun phrase that matches the number of the pronoun is considered as the referred phrase. The pronoun resolver begins the search for the noun phrase in the current sentence, and proceeds to the preceding sentences if a candidate noun phrase is not found in the current sentence. The resolver assumes the entities being referred to as being the subjects of the sentences, excepting sentences in passive voice. The heuristic followed is that the first noun phrase before the verb of a sentence is the subject of the sentence. The identified noun phrases are used to replace the pronoun in the sentence. The search for the noun phrase is also dependent on the voice of the verb. The noun phrases are identified in preceding sentences from the sentence with the pronoun for active voice. Sentences in which the verb group is followed by 'by' are marked as passive voice. In such cases, the candidate noun groups are those that match the number of the pronoun after 'by'. The sentences are processed sequentially in the order of occurrence in the abstract, only the immediate predecessor to the current sentence is considered if a corresponding noun group is not found in the current

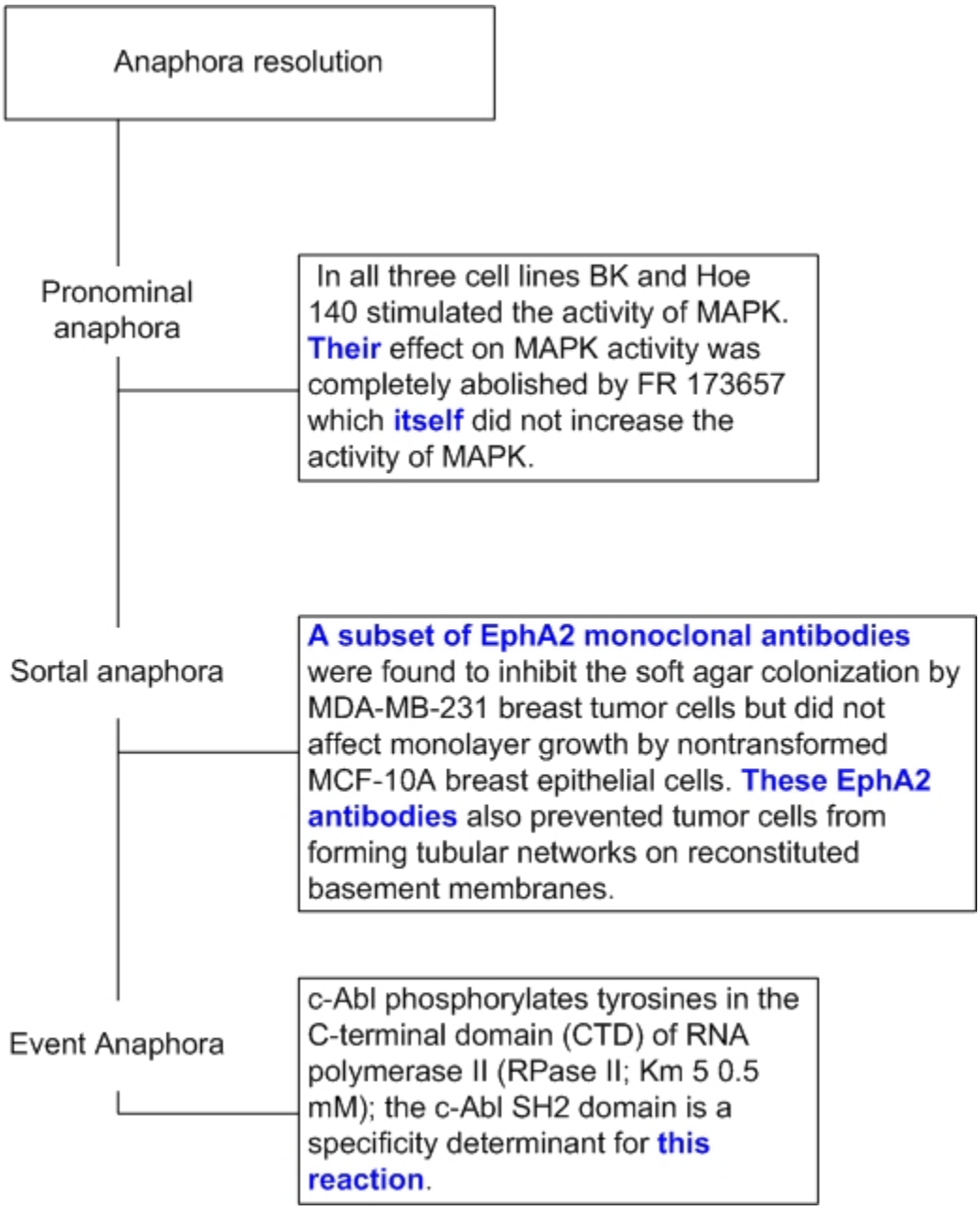

Figure 8. Illustration of Anaphora Types in Literature

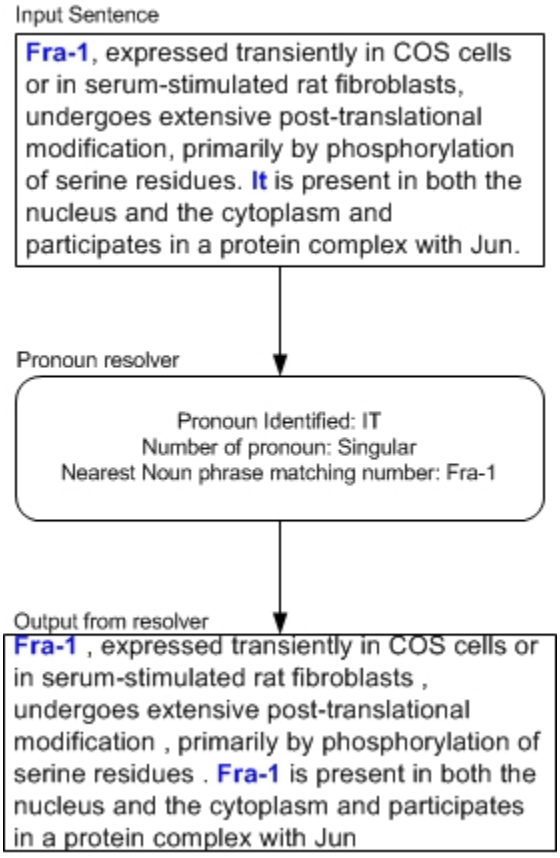

Figure 9. The Pronoun Resolution process

sentence. The cases handled by our system are restricted to a limited set of third person pronouns- it, itself, they, themselves, its and their. Sortal and event anaphora are on the cards for future work. A sample resolution is given in Figure 9.

2. Entity Tagging

The entity tagging system marks the names of gene, proteins, chemicals and cellular location in text. The process of tagging is a combination of dictionary look up and heuristics. Regular expressions are also used to mark the names that do not have a match in the dictionaries.

The dictionaries for the entity tagger are derived from various biological sources such as the Unified Medical Language System[1] (UMLS) from National Library of Medicine, Gene Ontology from the Gene Ontology Consortium[2] and National Center of Biotechnology Information's[3] (NCBI) Locuslink database.

The gene and protein name dictionary was extracted from LocusLink. The LocusLink database from NCBI provides descriptive information about the genetic loci of organisms. Official nomenclature, aliases, sequence accessions, phenotypes, EC numbers, MIM numbers, UniGene clusters, homology, map locations, and related web sites are among the information provided for the loci in various organisms. The LocusLink database is available as a flat file for download. A perl parser was written for the flat file that uses regular expressions to extract the filed values for the gene and protein names from the LocusLink records. The LL_tmpl file from the LocusLink ftp site was parsed and the values of the fields given in Figure 10 for all the organisms formed the gene and protein name dictionary. The dictionary has 508,477 entries corresponding to genes and proteins across different organisms.

A regular expression with 80% coverage on the extracted dictionary was also developed as a heuristic to mark the gene names not in the dictionary. The pattern addressed covers alphanumeric and hyphenated phrases as gene names, and excludes purely numeric or alphabetic names as possible gene names. Figure 10c shows the regular expression used.

Chemicals and cellular locations were also tagged in the text to identify modifier information such as the agents and sites of interaction between the genes and proteins. Dictionaries for these entities were derived from UMLS and GO. The UMLS knowledge

[1] http://www.nlm.nih.gov/research/umls/
[2] http://www.geneontology.org/
[3] http://www.ncbi.nlm.nih.gov/LocusLink/

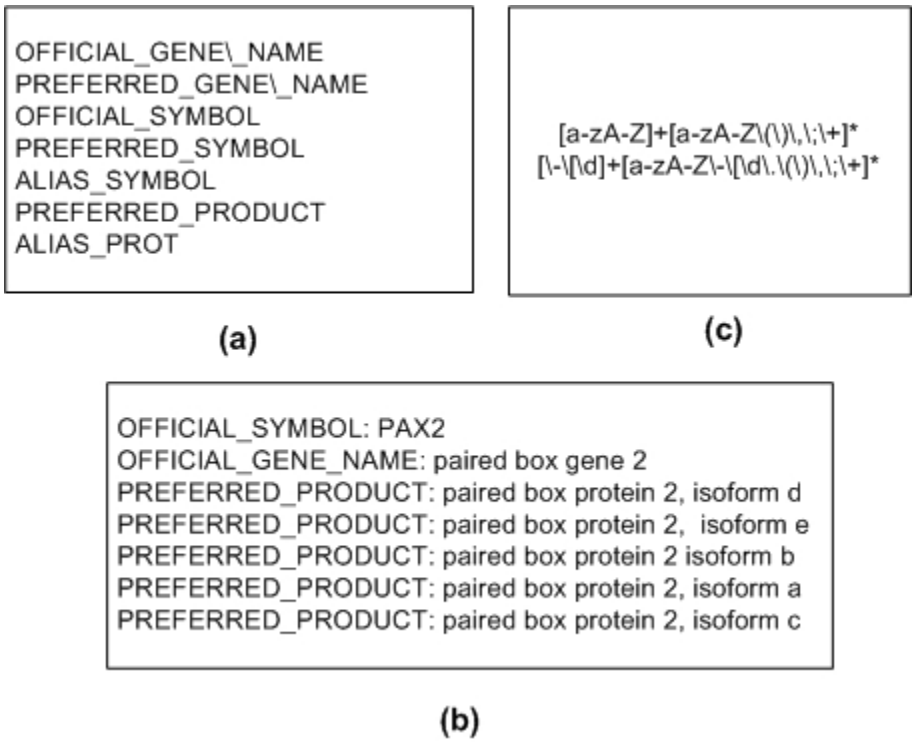

Figure 10. Tagging Gene and Protein Names (a) LocusLink fields considered for extraction (b) Partial LocusLink record (c) Regular expression used for gene name matching

sources consist of the Metathesaurus, the Semantic Network and the Specialist Lexicon. The Metathesaurus is a compiled vocabulary of biological terms, their synonyms and relationships between them. The vocabulary of Metathesaurus is derived from a combination of sources such as the MeSH, SNOWMED and various other biomedical dictionaries across different languages. The Semantic Network assigns the semantic categories to the words in the Metathesaurus and provides the relationships that might be assigned to the semantic types. The concepts in UMLS form the nodes in the Semantic network and the edges define the relationships between the concepts. Major groupings of semantic types for organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas are provided in the Semantic network. The Specialist Lexicon provides lexical information such as part of speech entries for the biomedical terms in the Metathesaurus as

well as commonly occurring English words. The Lexicon provides the information necessary for the SPECIALIST NLP system such as the syntactic, morphological and orthographic information for the terms. The UMLS knowledge sources are downloaded as tab separated files which form a highly interconnected set of tables in a database.

The dictionary for the chemicals was derived from the semantic types of *'chemicals'* and *'clinical drugs'* in the semantic network. The concepts and their synonyms corresponding to these semantic types are extracted from the Metathesaurus using a Java parser emulating a database table joins. The SRDEF file of Semantic network contains the association between the semantic type names and the semantic identifiers. The SRSTRE1 and SRSTRE2 files contain the hierarchy of the semantic network as relationships between the semantic types. The set of semantic types for chemicals corresponds to the sub tree under the semantic types of chemicals and clinical drugs. The sub tree was identified using the ISA relationship between the semantic types recursively. The UMLS MRSTY file parsed to identify the concept IDs corresponding to these semantic types IDs. The concept names and their synonyms were then extracted using the concept IDs from the MRCON file. The chemical names dictionary contains 790,000 entries corresponding to the chemicals recorded in the UMLS Metathesaurus.

The Location information dictionary was compiled in a similar way. The concepts and synonyms corresponding to the semantics types of *'cell', 'tissue', 'cell component'* and *'body part, organ/ organ component'* were extracted from the Metathesaurus. The Semantic network hierarchy of relations was used to identify the semantics types under the above mentioned semantic types. The concepts corresponding to these narrower types organized under the above mentioned roots were also added to the dictionary. In addition to the UMLS knowledge sources, the Gene Ontology was also used to add concepts to the Location

dictionary. The Gene Ontology (GO) is one of the controlled vocabularies of the Open Biological Ontologies and records information about genes and gene product attributes. The GO database contains three mutually exclusive ontologies describing the Molecular function of the genes, Biological processes that the genes participate in and the Cellular components that the genes are associated with in a species independent manner. Of these, the cellular components ontology has been used to enrich the Location dictionary of the extraction system. The Location dictionary contains 118, 018 terms describing the various cellular locations. Words and phrases in the text that match with entries in the dictionaries are tagged as the corresponding semantic entities using a prefix notation. The tagging format for the entities is given in Table 1.

Entity type	**Prefix**
Genes/Proteins	Gene
Chemicals	Chem
Location	Loc

Table 1. Tags used for Entities

An analysis of the dictionaries showed that entity names can also be numbers, alphabets and other common words such as 'be', ' was', 'A','686' etc. In order to avoid false positives as entity name matches, a list of stop words and POS tags were used. Any phrase that is a stop word is not considered a candidate for the entities. Sample stop words include the wh-words (which, where, when), connectives such as *and, yet, but*, punctuations and roman numerals. A set of part of speech tags of the words such as prepositions, connectives, adverbs and verbs were also used to eliminate words as candidate entity names. The phrases in the gene names were expanded by a process of relaxation - i.e., if a noun phrase contains a gene name, the entire noun phrase was marked as a single gene name entity. Hence *C-abl kinase activity* is marked as a gene name entity because *c-abl* in the noun group matches

an entry in the gene name dictionary.

3. Preprocessor

The tagged sentences need to be pre-processed to remove some constructs that cause the Link Grammar Parser to produce an incorrect output. The pre-processor sub-system also reduces processing time for the abstracts to certain extent by filtering out sentences that do not contain interactions. This section provides details on the working of the Pre-processor sub-system.

The sentences tagged with the entities are processed by the link grammar parser to extract the structural relationships. The word dictionaries of the link grammar parser are from conversational English which do not include the biological named entities. Further more, these entities do not follow English language norms and so the parser fails to produce correct parses, or, in some cases, fails to parse the sentence. This problem is overcome by forcing the parser to recognize the entity names as noun forms. Since the parser recognizes words that start with an uppercase letter as a noun, the multi-word entity names are converted to a single hyphenated word starting with an upper case letter.

The parser is also not designed for parentheses in the sentences. The sentences in the abstracts were analyzed, and it was found that the text inside parentheses often referred to alias names of the entities mentioned. So, the words in the parentheses were removed to improve the parse output as they provide no additional information in many sentences. However, there is some loss of information regarding the interactions due to this process which bring down the recall of the extraction system. Alternate approaches are being considered at this time.

The extraction process on an average takes about 75 seconds for an abstract. Hence, to reduce the processing time for an abstract, the sentences were further filtered, and those that contain at least two gene names and one interaction word were alone chosen for processing. This assumption is valid because the pronouns in the sentences have already been replaced by the corresponding noun groups. The interaction words are again identified using a dictionary built from biomedical databases. The UMLS Specialist Lexicon was used to identify the verbs commonly used in biological text. The set of verbs was then compared with the verbs in WordNet [30], the electronic lexical database. The verbs from Specialist that occur in WordNet were then eliminated from the list, under the assumption that these verbs are common English verbs. The verbs retained in the list refer to the verbs used purely in biological sense. The dictionary from these databases was also enriched manually with additional verbs that were known to refer to interactions. Since verbs, unlike nouns have different forms based on tense, number and person, a stemming algorithm, the Porter Stemmer [32] was used to normalize the words and match them against the list. The Porter Stemming algorithm is a suffix stripping algorithm that iteratively strips the verbs of their suffixes such as 'ed, 'ing','s' etc to arrive at the root form of the verb. As an illustration, the verbs 'running' and 'runs' are reduced to their root for of 'run'.

The pre-processor module also performs minor punctuation corrections on the spacing of commas and semi-colons in the text. The tagged and filtered sentences are stored in a bar separated file with the PubMed ID of the abstracts and the sentence numbers in order of occurrence in the abstract for back-reference. Figure 11 illustrates the tagging and preprocessing effects on sample sentences. These sentences are then processed by the link grammar parser to identify the simple sentences in the clause format used by the system. The following chapters explain this process in detail.

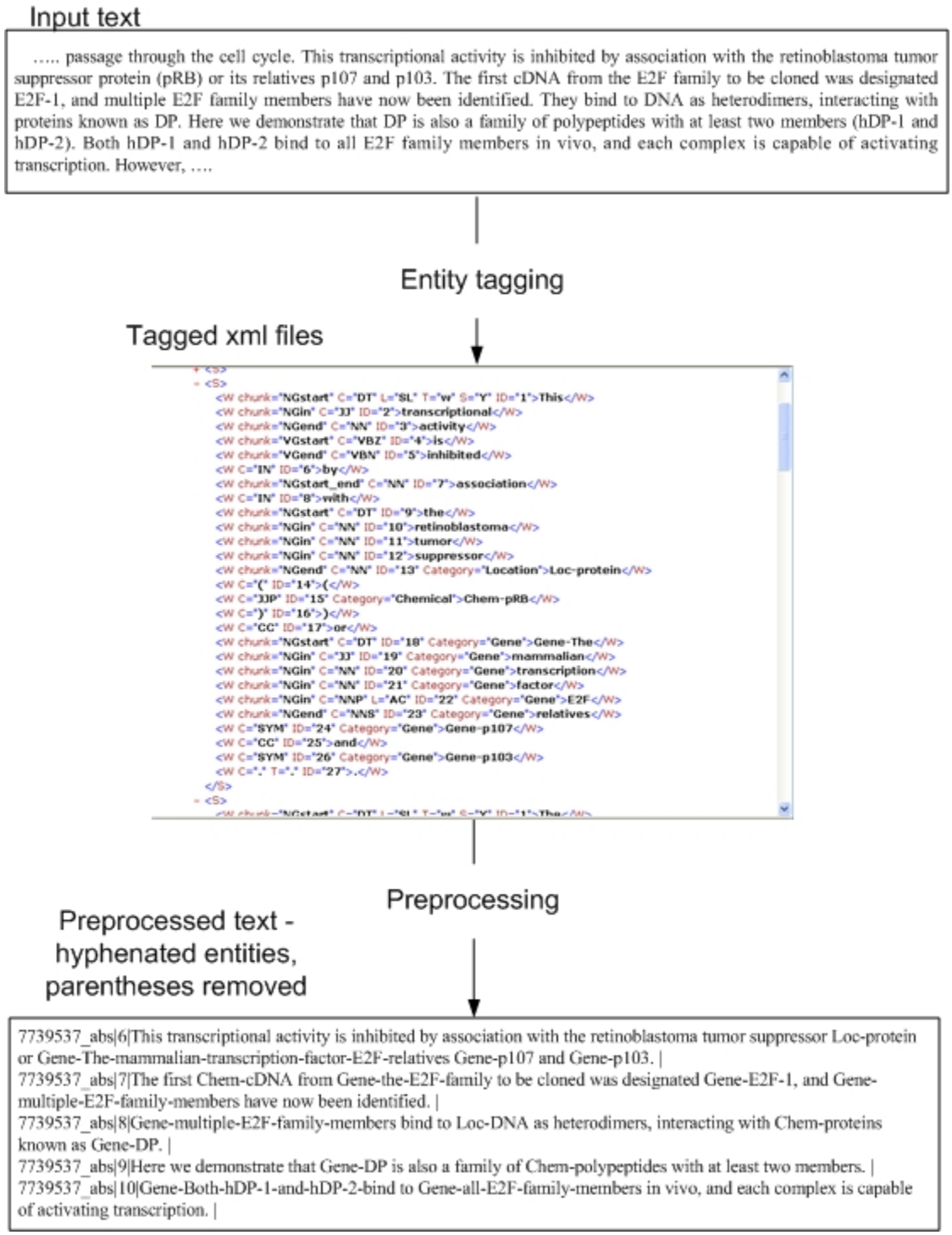

Figure 11. Illustration of Entity Tagging and Preprocessing

CHAPTER 6

Complex Sentence Processing using the Link Grammar Parser

The Complex Sentence Processor acts on the tagged and pre-processed text to produce the database of simple sentences. The sub-system uses the Link Grammar Parser (LGP) by Sleator and Temperly as the Natural Language Processor for this purpose. The LGP produces links between the words in a sentence that correspond to the syntactic structure of the sentence viz subject, object, determiner etc. The Complex Sentence Processor identifies specific links from the linkage output of the LGP and follows the links to obtain the syntactic constructs such as Subject, Verb, Object and Verb modifying phrase. The syntactic constructs are the representation of a simple sentence. The next section gives a brief overview of the Link Grammar Parser, and the processing of the linkages is outlined by the section following it.

1. The Link Grammar Parser

The Link grammar parser is based on link grammar, a syntactic dependency grammar of the English language. Grammars are a collection of rules that specify the allowed sentence structures in a language. Formal grammars can be either generative or analytical

in nature. Generative grammars use the specified abstract rules to produce sentences in the language addressed. Analytical grammar, on the other hand, provides an analysis of a given sentence's conformance to the rules. Analytical grammars for the English language have been represented using context free grammars, hidden markov models and probabilistic grammars, to name a few. Link grammars are a form of analytic grammar based on dependencies. In a dependency grammar, one word is the head of a sentence, and all other words are either a dependent of that word, or else dependent on some other word which connects to the head word through a series of dependencies. The Link grammar derives syntactic structure by examining the positional relationships between pairs of words[1]. The constructs (rules) in a link grammar represent the allowed connections between words in the sentence. A sentence in which all the words conform to their dependency connections is said to be in the language represented.

The Link Grammar Parser [34, 35] from Carnegie Mellon University is a syntactic parser written in C for the English language based on the Link Grammar. The parser is trained on words and sentences from telephone conversations on 70 different topics. A sample representation of a sentence in link grammar is given in Figure 12. The output of the link grammar for the same sentence is given in Figure 13. The parser contains word dictionaries that specify the possible part of speech assignments for the entries. A set of rules (linking requirements) are also written for each of the part of speech categories. These rules specifying the dependencies are represented using arcs between the words by the parser. The arcs connecting the words in the sentence are called as *links*. The links are directed, and are compatible only with the same links of different sign. Each word can have multiple links originating from it, like the word 'by' in the example. The word 'by' has a

[1]WordIQ Online Dictionary of Technical Terms, http://www.wordiq.com/

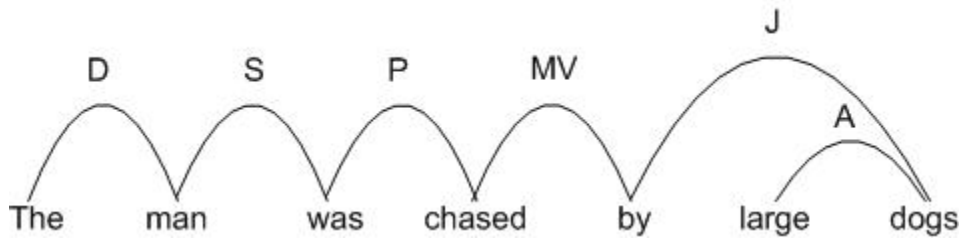

Figure 12. Link Grammar Representation of a Sentence

preposition connector link 'J' to the word 'dogs' and is connected as a modifier to the verb through the 'MV' link. The linking requirements of the words are represented using the Boolean connectors of '&' and 'or' in the dictionary. For example, the linking requirement of the word 'the' is given in the dictionary as

```
the: ({AL-} & {@L+} & (D+ or DD+)) or DG+ or (TR- & U+);
```

The links required to the right of the word are suffixed with a '+' sign, and those that are required to be on the left of the word are suffixed by '-'. Hence, in the example, D+ specifies that there should be a determiner link to the right of the current word. i.e., the current word should be followed by another word that has a D- in it linking requirement such that the link does not cross any other link between the words in the sentence. The parser tries to come up with an allocation of links for the words in a sentence. The links are constructed in a way that best satisfies the linking requirements of all the words in the sentence. A sentence is said to be in the language of the parser if the links satisfy the linking requirements of the words and follow the connectivity and planarity properties of the parser. The connectivity property states that all the words in the sentence should be connected to each other in some indirect way. The planarity property states that the links should not cross. A valid parse of the sentence is called as *linkage*.

Sentences in the English language often convey multiple meanings due to the ambi-

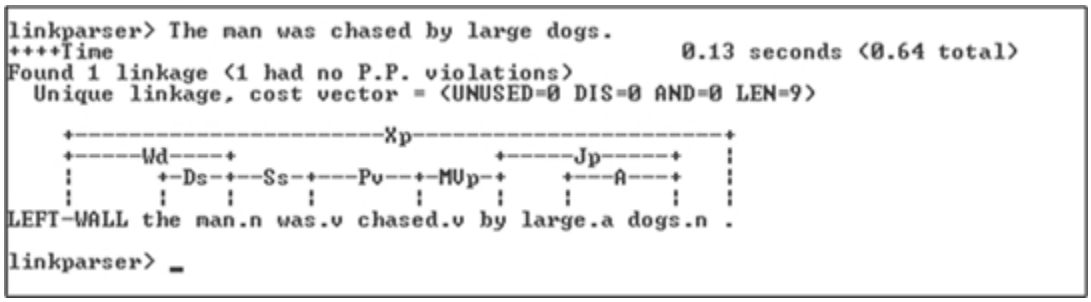

Figure 13. Link Grammar Parser's Output for the Example

guity of the semantics of the words in the sentence. Hence, as a reflection of this nature, the link grammar parser produces numerous linkages for a given sentence, based on the various combinations of linking requirements satisfied. The linkages are assigned costs and are displayed in increasing order of cost. Sentences with connectives tend to produce crossing links between the words Figure 14. In case of sentences with connectives, the parser produces independent sub-linkages focused on each clause connected by the connective. The *union* option of the parser allows the user to combine all the sub-linkages in a single linkage. The combined linkage has crossing links between the words, which is a special case of parsing. The parser is thus able to handle complex sentences robustly.

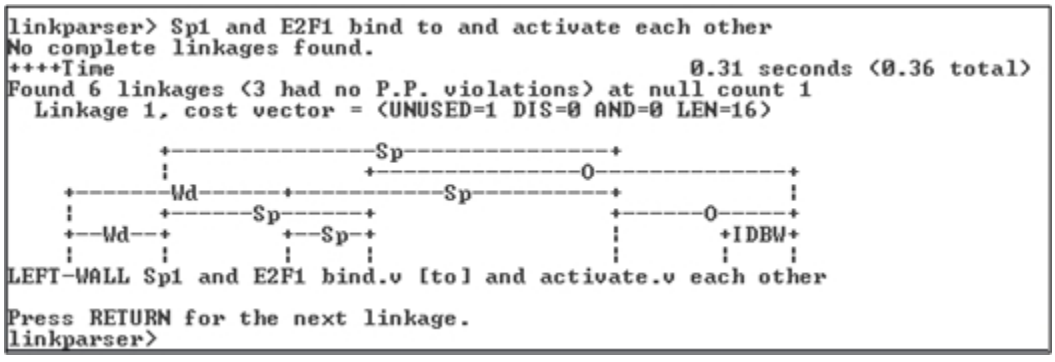

Figure 14. Sentences with connectives: Crossing links

The parser allows for customizing its behavior as per user requirements through *options* set by variables. The 'union' option combines the various sub-linkages produced in

the case of sentences with co-coordinating conjunctions. The complex sentence processor uses the union linkage to combine all possible sub-linkages to get a consolidated representation of the sentence structure. The *timeout* variable specifies the maximal time limit to be taken by the parser to assign links to the words. If the parser is not able to assign linkages completely satisfying the requirements within the time, it tries to neglect words and re-assign the linkages, allowing null linkages for certain words. Failing this, the parser enters the *panic mode* where the parser tries to assume the linking requirements of certain unknown or ambiguous words and assign a plausible linkage for the sentence. The memory required for parsing can also be set using the *memory* variable. The parser also allows for batch processing of sentences using the *batch* variable. A recent paper by Pyysalo et al [29] analyses the usage of link grammar parser for the biological domain. They suggest various methods to improve the parsing of sentences using the link grammar. Szolovits [36] also proposes a way to improve the parsing of biomedical sentences by adding words from the UMLS Specialist Lexicon to the dictionaries of the link grammar parser.

We have used the Link Grammar Parser for our purpose as it provides the syntactic constructs that relate to the linguistic rules for a sentence. Hence, by using the parser, our approach is linguistically oriented.

2. Complex Sentence Processor

The complex sentence processor splits the complex sentences into the internal clause format representing simple sentences using the link grammar parser. The link grammar parser is operated in union mode to get the combined linkage of all sub-linkages. The union option of the parser was set and the time out variable was set to 30 seconds. As a result,

if the parser is unable to construct a linkage for the sentences in less than 30 seconds, it produces a probable linkage, ignoring some words and assuming the roles of others.

The parser used currently is the windows version of the link grammar system distributed by Sleator et al. It was found that the parser is not able to handle sentences of more than 70 words. There is also a restriction on the length of the words in the sentences, making the parser crash if the words are more than 50 characters in length. Sentences are further filtered based on these criteria and only those conforming to the limits are sent to the link grammar parser. The linkages from the parser are then sent to the Complex sentence processor for analysis. The complex sentence processor currently analyzes the first linkage returned by the parser. Since the linkages are returned in increasing order of the cost for constructing the linkages, the first linkage is most often the best linkage possible for the sentence.

The Complex sentence processor follows a verb-based approach to extract the simple sentences. The assumption is that the verb represents the central idea of a clause. A sentence is identified as complex if there is more than one verb in the sentence. The links from the verbs are followed to get the subject, object and modifying phrase. The clause format used to represent simple sentences is given below.

```
Subject + Verb + Object + Modifying phrase to the verb
```

The modifying phrases to the verb can be adverbial, prepositional or adjectival phrases. The components can be a single word or multi-word phrases. Each of the components, once identified is expanded to include multi-word phrases. The links used to expand the words to phrases are the determiners, adjectives or prepositional attachments. Hence subjects like 'The kinase activity of C-abl' can also be extracted. The algorithm followed

for complex sentence processing is given in Algorithm 1. A trace of the process for a sample sentence is given in Figure 17. A more detailed trace of the working of complex sentence processing algorithm is given in Figure 15. Figure 16 provides a brief summary of some of the links used by the algorithm.

Since there are no pronouns in the example, the sentence does not undergo any change when passed through the pronoun resolution module. *p107 wee 1 Tyr-15* and *p34cdc2* are tagged as gene names by the entity tagger. The pre-processing module identifies that the sentence had two gene names and an interaction word (phosphorylating) in the sentence. The interaction word 'phosphorylating' stems to 'phosphorylat' and matches the stemmed entry in the interaction words dictionary. Hence the sentence passes the filter and is sent to the link grammar parser. The first linkage obtained for the sentence is also shown in the figure. The linkage obtained shows three clauses as part of the sentence, as indicated by the three S links from the verbs 'indicate', 'functions' and 'is'. The subjects, objects and modifying phrases for each of these verbs are identified as per the algorithm. The final output of the complex sentence processor is a set of three clauses for the sentence as seen in the Figure 17. Figures 18 and 19 illustrate the working of the extraction system for sentences for a few other sentences. The example in Figure 18 illustrates the working of the Complex Sentence Processor when references to entities in the absence of pronouns are encountered. The entity *CD13/aminopeptidase N* is referred implicitly by the presence of 'which', without the usage of pronouns. The sentence is also in passive voice, and the interaction extracted *DMP1 # activated # CD13/aminopeptidase-N#* takes care of the passive voice and extracts the directionality of the interaction precisely. Coordinating conjunctions are also taken care of, as shown in the example. Figure 19 shows the working of the system for sentences with pronouns and multiple themes in the sentence (independently isolated

Algorithm 1 Algorithm for Complex Sentence Processor

```
1. Identify sentences to process:
    a. If number of S links > 1, go to step 2.
2. Processing complex sentences - Set ranges in sentence
    a. Identify the S links in the sentence.
       Each S link forms the beginning of a simple sentence.
    b. The word to the left of the S link forms the initial
       starting and ending range of the subject
        i. If the subject word is 'which', follow the MX link from 'which'
           to obtain the word that 'which' is pointing to.
           The subject ranges correspond to the pointed word.
       ii. Obtain the determiners, proper nouns, prepositional and
           adjective phrases occurring to the left of the subject.
      iii. Append subject-related adjective, adverb and prepositional
           phrases following the subject.
       iv. Follow the CO link to the left of the subject to get the
           subject modifying phrase.
    c. The word to the right of the S link forms the starting and
       ending point of the verb range. Expand the range of the verb as below.
        i. Obtain the infinitives, adverbs and verb complements
           connected to the main verb.
       ii. Follow the N links from the main verb to get
           negative forms of the verb.
    d. Follow the O links starting from the end of the verb range to get
       the initial range of the objects.
       For each object, expand ranges as below
        i. Obtain the determiners, proper nouns, prepositional and
           adjective phrases occurring to the left of the object.
       ii. Obtain the determiners, proper nouns, prepositional and
           adjective phrases occurring to the right of the object.
    e. For each modifying phrase connected to the verb, do
        i. Expand range to include the prepositional , adjective and
           adverbial phrases occurring to the right of the modifier link.
       ii. Follow the O links which are not connected to the main verb to include
           the objects of adverbial phrases and gerunds.
3. Extract based on ranges - Extract the components of the simple sentence
    from the original sentence using the ranges as the substring markers.
```

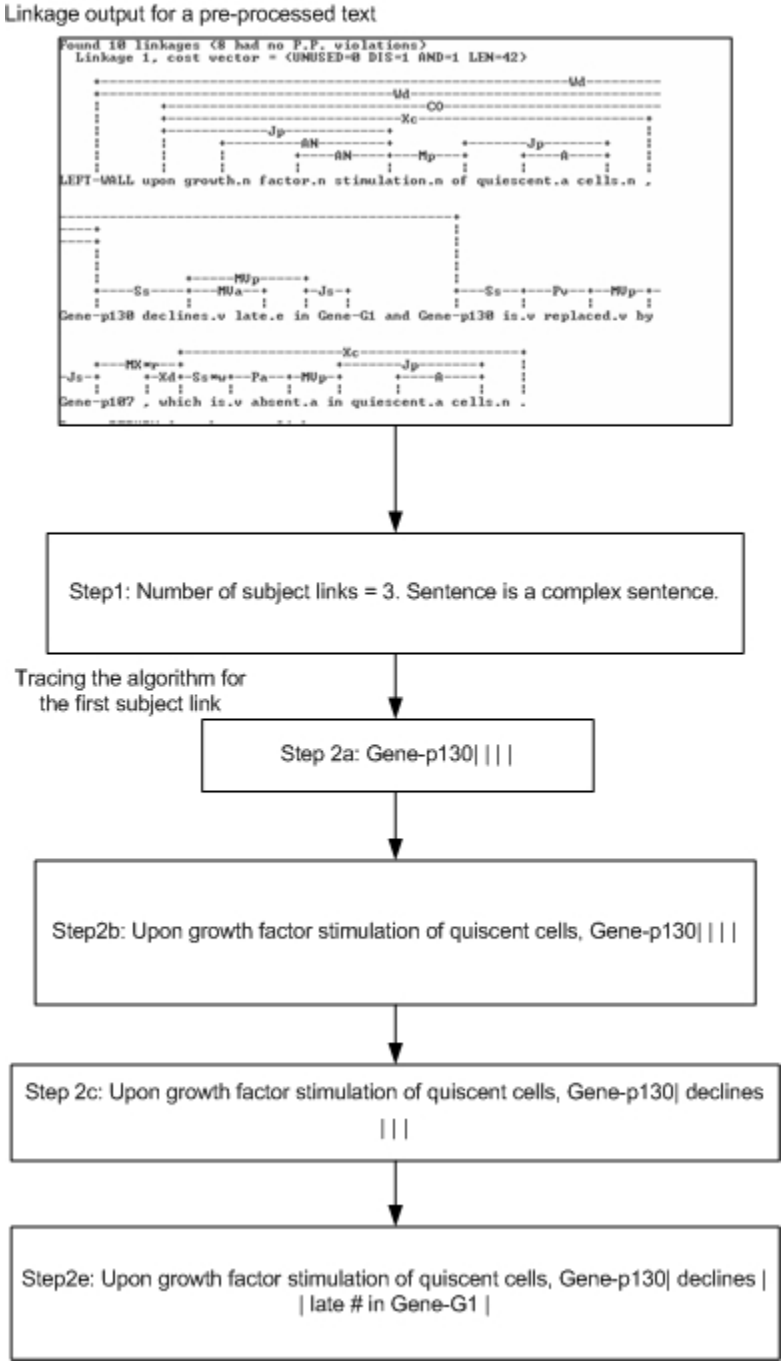

Figure 15. Trace of the Complex Sentence Processing Algorithm

A	connects pre-noun ("attributive") adjectives to following nouns: "The BIG DOG chased me", "The BIG BLACK UGLY DOG chased me".
B	serves various functions involving relative clauses and questions. It connects transitive verbs back to their objects in relative clauses, questions, and indirect questions ("The DOG we CHASED", "WHO did you SEE?"); it also connects the main noun to the finite verb in subject-type relative clauses ("The DOG who CHASED me was black").
CO	connects "openers" to subjects of clauses: "APPARENTLY / ON Tuesday , THEY went to a movie".
D	connects determiners to nouns: "THE DOG chased A CAT and SOME BIRDS".
E	is used for verb-modifying adverbs which precede the verb: "He is APPARENTLY LEAVING".
G	connects proper noun words together in series: "GEORGE HERBERT WALKER BUSH is here."
I	connects infinitive verb forms to certain words such as modal verbs and "to": "You MUST DO it", "I want TO DO it".
J	connects prepositions to their objects: "The man WITH the HAT is here".
MV	connects verbs and adjectives to modifying phrases that follow, like adverbs ("The dog RAN QUICKLY"), prepositional phrases ("The dog RAN IN the yard"), subordinating conjunctions ("He LEFT WHEN he saw me"), comparatives, participle phrases with commas, and other things.
MX	connects modifying phrases with commas to preceding nouns: "The DOG, a POODLE, was black". "JOHN, IN a black suit, looked great".
N	connects the word "not" to preceding auxiliaries: "He DID NOT go".
O	connects transitive verbs to their objects, direct or indirect: "She SAW ME", "I GAVE HIM the BOOK".
P	connects forms of the verb "be" to various words that can be its complements: prepositions, adjectives, and passive and progressive participles: "He WAS [ANGRY / IN the yard / CHOSEN / RUNNING]"
S	connects subject nouns to finite verbs: "The DOG CHASED the cat": "The DOG [IS chasing / HAS chased / WILL chase] the cat".

Figure 16. Explanation of Links Used by the Algorithm

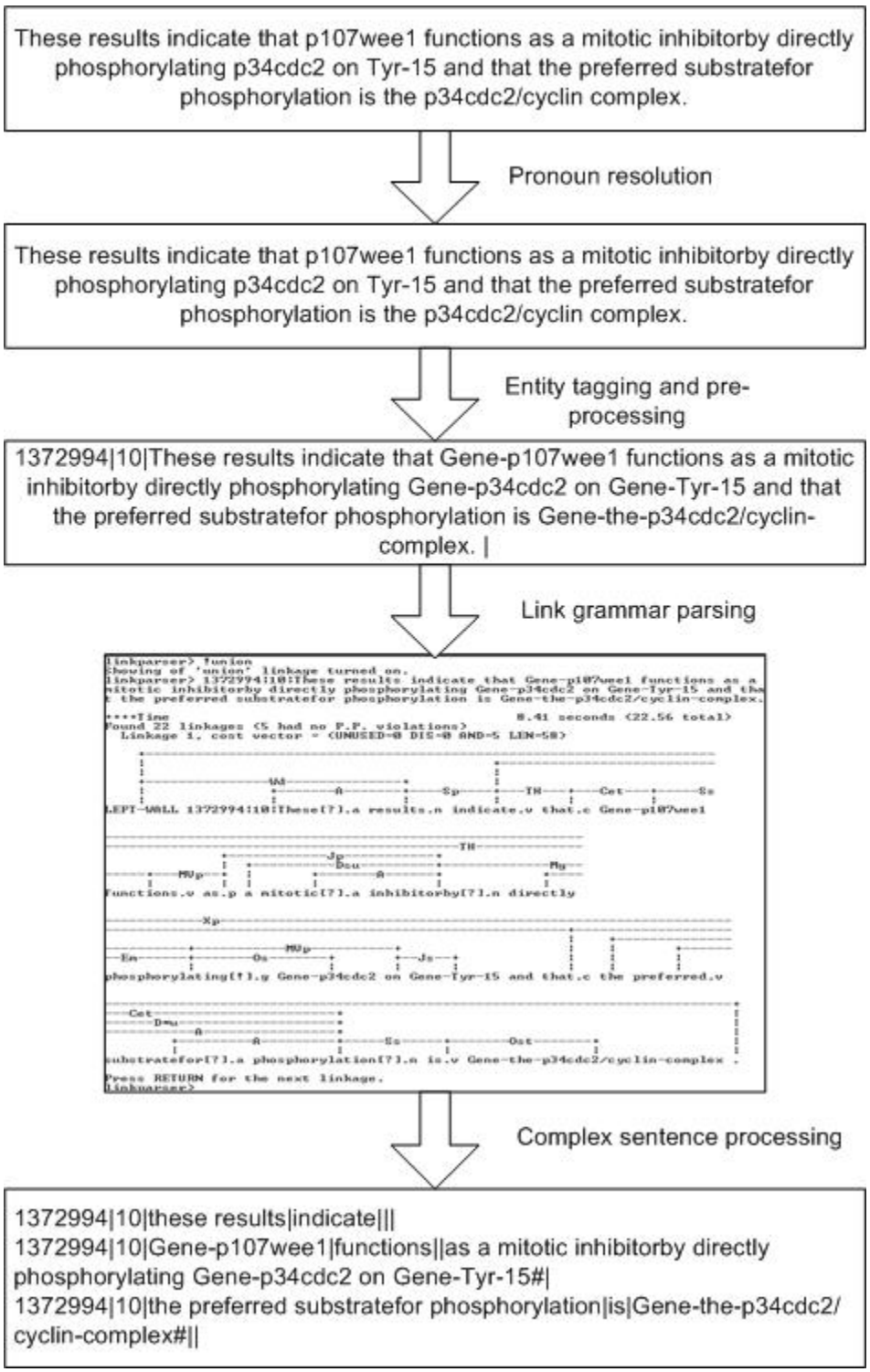

Figure 17. Complex Sentence Processing: an example

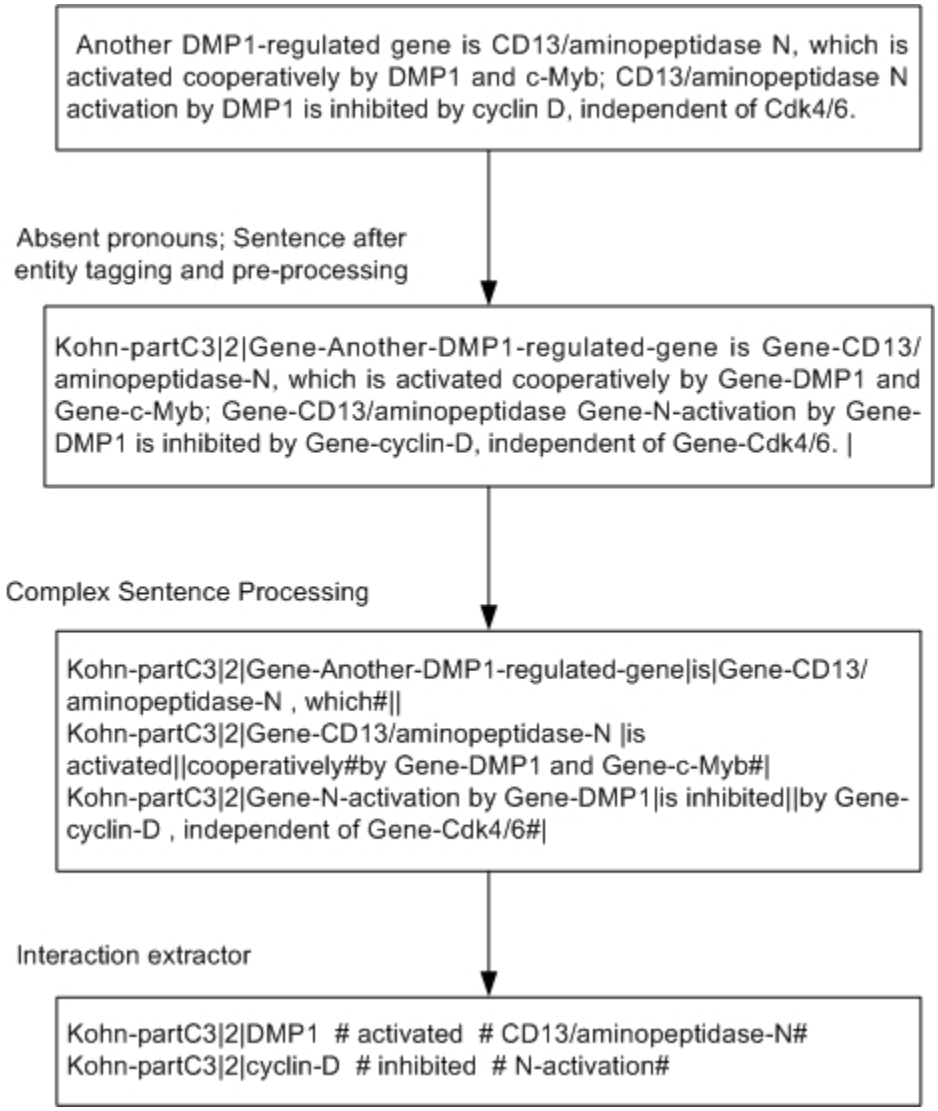

Figure 18. Interaction Extraction from Complex sentences: Passive voice, coordinating conjunctions and reference in the absence of pronouns

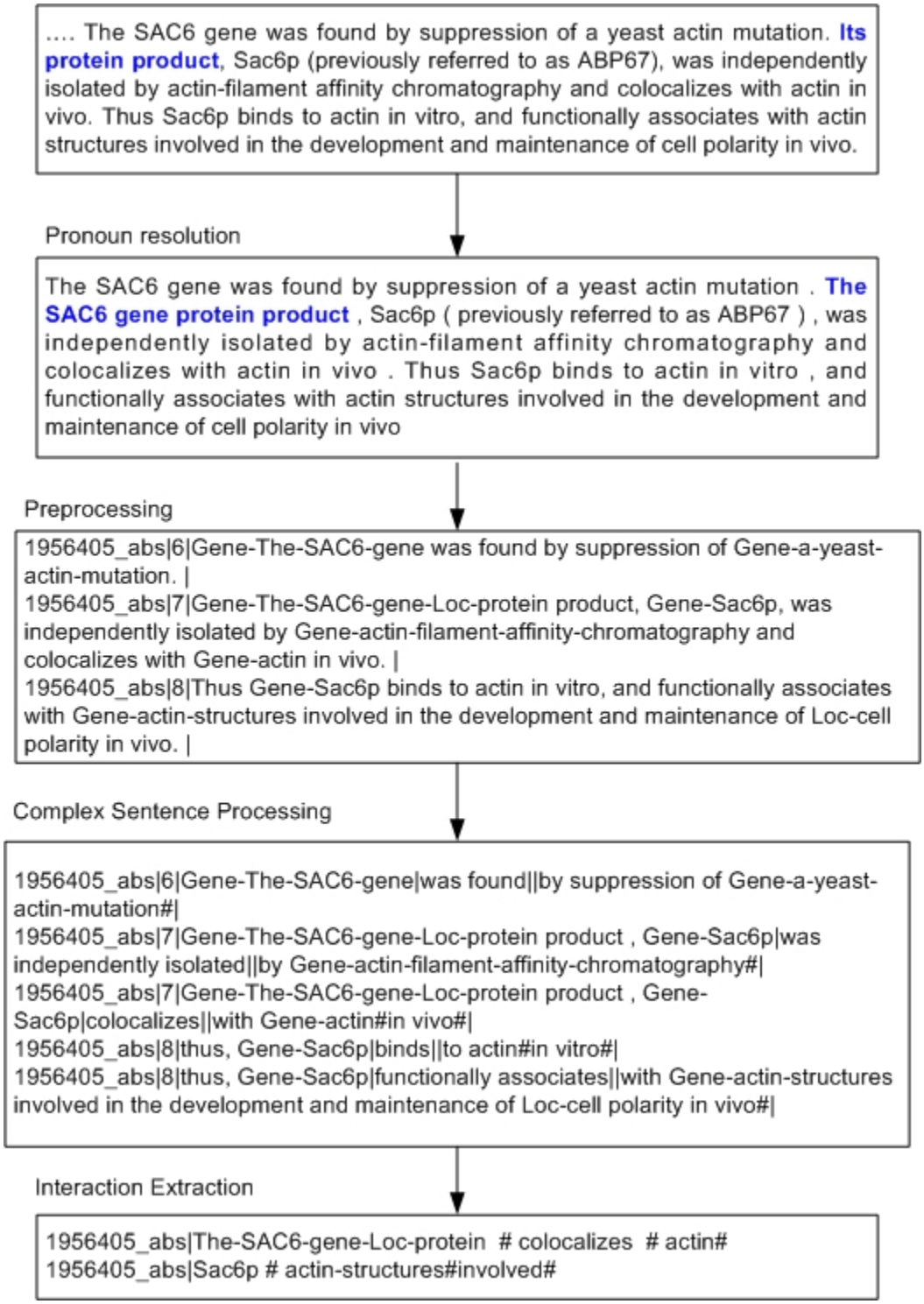

Figure 19. Interaction Extraction from Complex sentences: Pronoun resolution

by actin-filament affinity chromatography and colocalizes with actin in vivo).

The complex sentence processor was tested on 66 abstracts from the breast cancer domain and was shown to perform well most of the time in identifying the simple sentences if a correct parse of the sentence is obtained. However, the complex sentence processor fails to produce the correct representations of simple clauses in sentences with negative conjunctions. As an example, consider the sentence below and the output of the complex sentence processor given in Figure 20. This is because the conjunctions 'neither...nor' are not recognized by the link grammar. The link grammar parser ignores the conjunctions in a sentence when trying to assign linkages to a sentence with connectives such as co-coordinating conjunctions. There are no links connected to these words from the other words. Hence, the meaning of the sentence is lost since these words cannot be included in constructing the clauses. The relationship between the clauses is also compromised. This problem can be overcome if the phrase boundaries are checked for un-linked words and extended to include semantically significant words that are not linked to other words. An analysis of the connectives might also lead to identify pathways in text as some connectives such as 'when', 'and therefore' tend to specify causal relationships between the clauses.

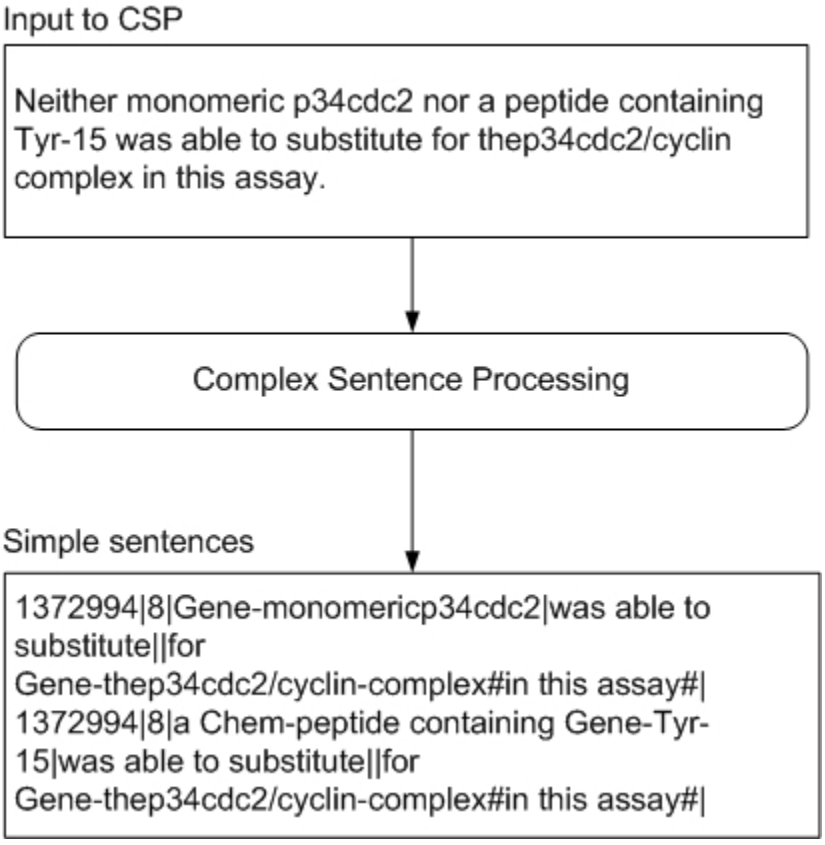

Figure 20. Negation in Complex Sentence Processing

CHAPTER 7

Evaluation

Current information extraction systems have been evaluated on the basis of three measures: precision, recall and f-measure. Precision is a measure of correctness of the system, and is calculated as the ration of true positives to the sum of true positives and false positives. The sensitivity of the system is given by the recall measure, calculated as the ratio of true positives to the sum of true positives and false negatives. F- Measure is a combination of Precision and Recall and provides an estimate of the system performance in a single measure. Considered as the harmonic mean of Precision and Recall measures, F-measure is given in Equation 7.1.

$$F - measure = \frac{2 \times P \times R}{P + R} \tag{7.1}$$

.

Our extraction system is also evaluated on these measures. Precision and Recall measures are calculated for our system as given in Equations 7.2 and 7.3.

$$Precision = \frac{No.\,of\,interactions\,extracted\,correctly}{No.\,of\,interactions\,extracted} \tag{7.2}$$

$$Recall = \frac{No.\,of\,interactions\,extracted\,correctly}{No.\,of\,interactions\,present\,in\,text} \tag{7.3}$$

The Precision value of a system is a measure of the specificity of the system. It gives an idea of the correctness of the system by measuring the number of times the results are extracted correctly in comparison with the total number of results. Recall is a measure of sensitivity of the system, giving an account of how often the system is able to extract the right results.

The complex sentence processing module is a part of the automated information extraction system for biomedical text. This chapter focuses on the performance of the extraction system in comparison with other existing systems. The module presented in this thesis takes care of both entity tagging and natural language processing of the text in addition to identifying simple sentences. Since a comparable system for complex sentence processing is not available in the domain for analysis, the results of the module are presented in terms of the results of the extraction process to facilitate comparison with approaches in use.

Among the related systems, Ono et al [11] used manually defined set of pattern matching and part-of-speech rules to extract interactions for only four keywords 'interact', 'associate', 'bind', 'complex' and their inflections and have achieved precision of 94.3% and a recall of 86.8% on yeast interactions and a precision and a recall of 93.5% and 82.5% on E.coli interactions on selected sentences. MedScan [14] uses protein name dictionaries and transition networks to achieve a coverage rate of 43%. Incorporating protein-function ontologies into their system, MedScan is able to achieve results with a precision of 91% and a recall of 21% of the interactions between human proteins on MEDLINE abstracts dated after 1988. GENIES [15] , by the use of a tagger and pattern matcher is able to

achieve a precision of 96% and a recall of 63% for binary relations and a precision of 88% for nested relations. The system results are however on a single full text article from the Cell magazine.

Context free grammars used by Temkin [20] provide another direction in rule engineering. The system achieves a precision and recall of 70% and 64% on 100 randomly selected abstracts from PubMed. Our extraction process focuses on whole sentences, while their grammar focuses on syntax of prepositions, protein names and transitive words. Leroy and Chen [21] use preposition based templates, creating an automata like approach to extracting interactions. They engineer rules for 2 prepositions, with a precision of 70% on 50 abstracts.

The Nave Bayes classifier approach of [23] gives a precision measure of 92% and a recall of 21% on abstracts from YPD, using weakly labeled training instances. Our approach using link grammar parser automates the process of extraction of interactions, limiting the need of hand-built patterns to a small scope of sentence constituent (e.g. subject) instead of the whole sentence.

We have evaluated the performance of our system against two existing systems - BioRAT [17] and GeneWays citegeneways. The BioRAT system uses manually generated pattern rules and dictionaries (GATE gazetteers) of entity names for their extraction system. Natural language processing is done using the GATE [18] tool. The authors also evaluated the BioRAT system against Blaschke's SUISEKI [10, 13] system. The evaluation against SUSEIKI was performed on abstracts chosen from the protein interaction records in DIP [37], the protein interaction database. The DIP database contains interactions between proteins mentioned in articles curated by experts and extracted using an information processing system. The authors chose 389 interactions from DIP in which both the proteins

participating in the interaction had SwissProt entries. These interactions correspond to 229 abstracts from PubMed. The BioRAT system was evaluated against these 229 abstracts and the authors report a recall of 20.31% and a precision of 55.07% on the interactions extracted from their system. The recall of BioRAT was comparable to the recall of the SUISEKI system (22%).

We evaluated our extraction system also against the 229 abstracts[1] from the BioRAT evaluation module. The interactions extracted by the system were then manually examined for precision and recall. The DIP database was used as the standard for evaluation of our results. The database is available for download as an XML file. The DIP XML file represents proteins as nodes and the interactions between the proteins as edges between the nodes. The nodes are annotated with their identifiers from various databases such as SwissProt, NCBI and HUGO. The XML file of DIP was processed to create a tab delimited file of interactions, indexed by the PubMed ID of the article citing the interaction. The SwissProt database was also downloaded as an XML file, and was parsed to obtain the names and aliases of proteins mentioned in SwissProt. The DIP entries were augmented with the alias list from SwissProt using the SwissProt IDs of the nodes. Our protein name dictionary was also limited to the SwissProt entries, so as to provide a fair comparison with BioRAT.

The precision of the system was calculated by manually comparing the extracted interactions against the abstracts. The interactions specified in the abstracts were marked were marked as correct or incomplete based on the nature of match. Exact matches of the interactions extracted and the text score a correct extraction. Incomplete extractions are those in which the interactions bear a complete meaning only by the addition of prepositional

[1]Data obtained from David Corney by personal communication

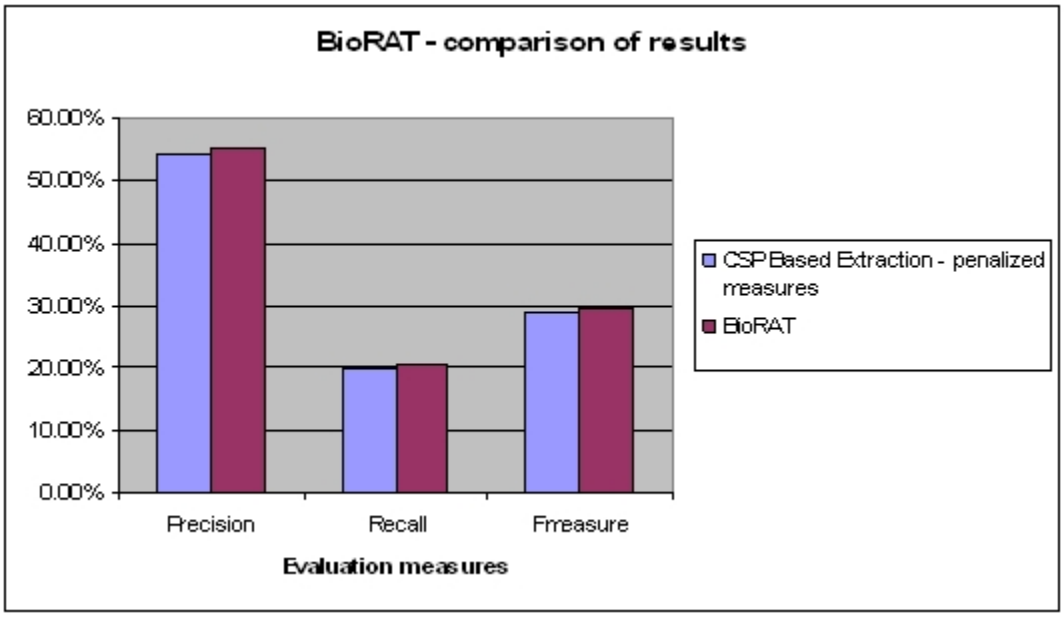

Figure 21. Comparison of results with the BioRAT system

or adjectival phrases to the participants. Our extraction system was able to identify 452 interactions from the 229 abstracts, with a precision of 54.39% and a recall of 19.94%. The F-measure of the system was 29.18 %. A comparison of our performance against the performance of the BioRAT system is given in Figure 21. Most of the errors were found to be due to the protein name tagging (36 %) and interaction extractor (20 %). The evaluation process against the BioRAT dataset was a team work, with contribution by Syed Ahmed. An analysis of the reasons for the drop in precision and recall is given in Figure 22.

Since our extraction system was tested on abstracts, we tend to miss out on some interactions present in the full text that have been curated by the experts. So, we conducted an analysis of the interactions missed out due to this reason and the results show that only about 4% of the interactions were missed because they were not present in the abstract. Figure 23 shows the effect of the full text on precision, recall and f-measure of the system on the 229 abstracts.

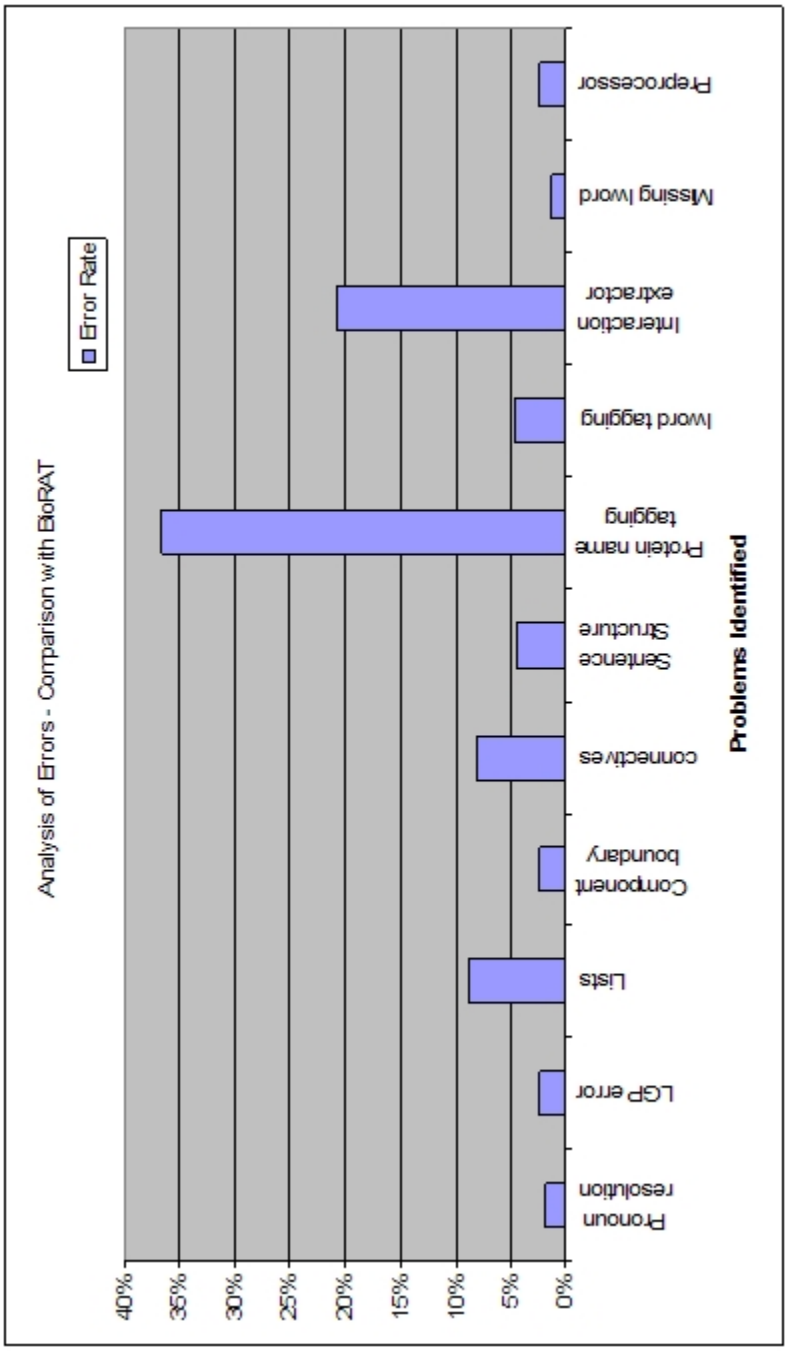

Figure 22. Analysis of Errors in Precision and Recall - BioRAT

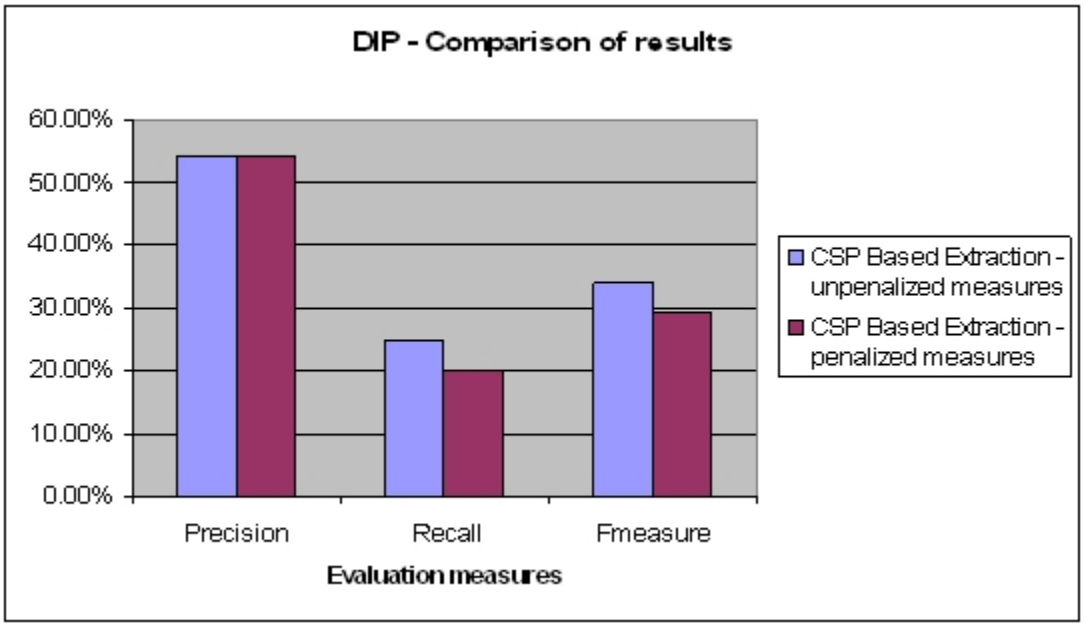

Figure 23. Interactions missed in abstracts

We have also evaluated our system against the GeneWays system [16]. The GeneWays system is based on the GENIES text extraction system that uses hand written semantic rules to extract gene interactions from text. We were able to obtain the results from GeneWays on 50,000 articles showing the calculation of precision on the dataset[2]. Of the interactions output by the system, the authors identified 2500 unique interactions that had more than 10 citations in the articles and evaluated their precision with the help of a domain expert. The GeneWays system reported a precision and recall of 95 % and 65 % respectively. We also tried to replicate their process of evaluation on a dataset of 495 abstracts queried from pubmed, corresponding to the 6 most cited interactions extracted by GeneWays. The abstracts were queried from PubMed using the eutilities facility of NCBI using the participants of the interactions as keywords. Each interaction resulted in 20 abstracts. The abstracts, when run on our extraction system resulted in 1600 interactions, of

[2]Data obtained by personal communication with Dr. Andrew Rzhetsky

which we calculated our precision on 156 interactions that occurred more than twice in the abstracts. Our system extracted interactions with a precision of 76.13%. The recall of the GeneWays system was represented by the recall analysis on GENIES, the text extraction component of GeneWays. GENIES was tested on a full text article from Cell [38]. We ran our extraction system on the full text article and obtained 42 interactions with a recall measure of 63.64 %. Analysis of errors in precision shows that a majority of the errors in the extraction process were due to bugs in the interaction extractor module and identification of interaction words and proteins. Figure 24 shows the contribution of the problems identified in the system to the fall in precision. The comparison with GeneWays was an individual effort done by the author.

We also evaluated the manual system against the two data sets. The rule engineering system tested on the 495 abstracts from GeneWays interactions gave a precision of 69.57% against the 76.13% we got from the automated system. The interactions which occurred at least twice in the results were filtered from analysis as in the comparison of the automated system. The full text article used for recall analysis gave a recall of 27.27%. The system was also compared with the BioRAT dataset for 50 abstracts and the precision and recall measures for the 260 interactions evaluated were 30% and 23.07%. The F-measure for the system is 26.09%.

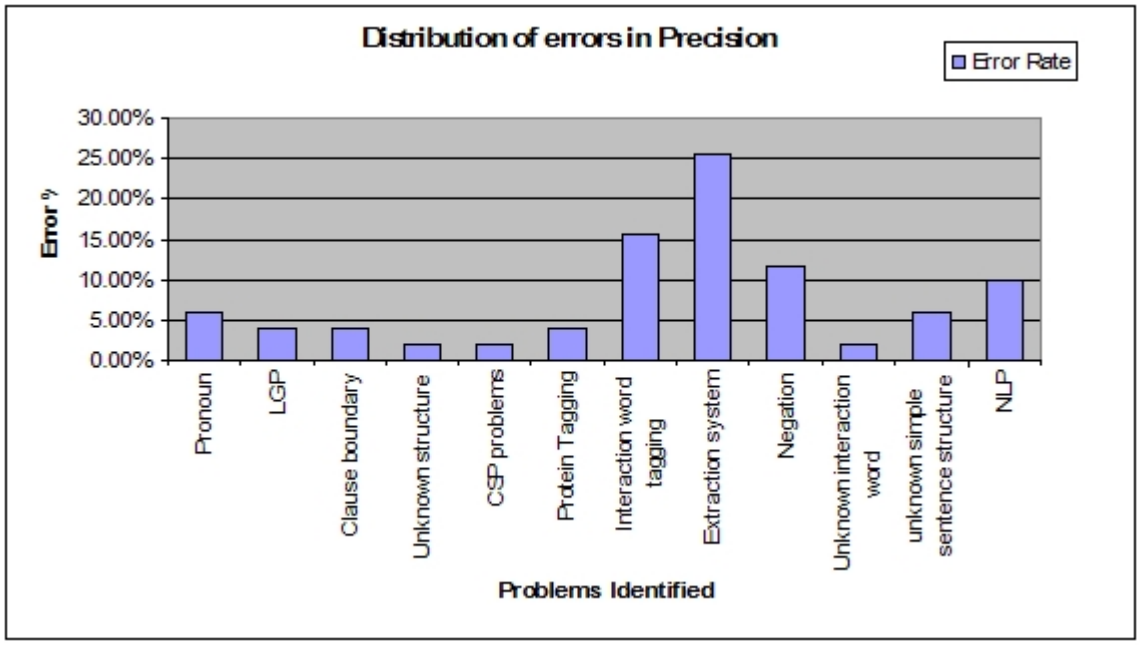

Figure 24. Analysis of Errors in Precision - GeneWays

CHAPTER 8

Conclusion

The thesis presented two approaches to extracting information based on sentence structures. While the manual approach is more accurate and can be engineered in a domain specific way, the automated approach is also advantageous because of its scalability. Our evaluation of the automated system on both the BioRAT and GeneWays datasets shows that our system performs comparably with other existing systems. Both the systems compared were built by manual rule engineering approach, and involve a repetitive process of improving the rules which take up a lot of effort and time. Our system is able to achieve similar results with minimal effort on part of the developer and user. While advantageous on this aspect, we realize that our system is also in need of improvements in tagging entities to boost the performance. Improvements in the interaction extractor module will also bring up the precision of the system. Nevertheless, we have proven that a syntactic analysis of the sentence structure from full sentence parses produces results comparable to many of the existing systems for interaction extraction.

1. Future Work

The information extraction system based on complex sentence processing is able to handle binary relations between genes and proteins, and some nested relations. However, researchers are also interested in contextual information such as the location and agents for the interaction and the signaling pathways of which these interactions are a part. Our tasks for future work include the following.

- Handling negations in the sentences (such as "not interact", "fails to induce", "does not inhibit")
- Identification of relationships among interactions extracted from a collection of simple sentences (such as one interaction stimulating or inhibiting another)
- Extraction of detailed contextual attributes (such as bio-chemical context or location) of interactions and
- Building a corpus of biomedical abstracts and extracted interactions that might serve as a benchmark for related extraction systems.

Attempts to improve the parse output of the Link Grammar System were also undertaken. The dictionaries of the Link Grammar Parser were augmented with medical terms with their linking requirements provided by Szolovits [36] in his website. In spite of the improvement in performance of the Link Grammar Parser, this approach was discontinued in favor of the Pre-processor subsystem because of the increase in time taken to load the dictionaries and for parsing.

Semantic analysis based on proposed information extraction techniques would enable automated extraction of detailed gene-gene relationships, their contextual attributes and

potentially an entire history of possibly contradictory sub-pathway theories from biomedical abstracts in PubMed thus allowing our system to generate more relevant and detailed recommendations.

In summary, the thesis provided a brief overview of the current trends and challenges in Chapter 2 and a pilot attempt at processing the complex sentences using a rule engineering approach was detailed in Chapter 3. Chapters 5 and 6 explained the various sub-systems of the Complex Sentence Processor. The evaluation of our system was detailed in Chapter 7.

REFERENCES

[1] Cohen K. Bretonel and Lawrence Hunter. Natural language processing and system biology, 2004.

[2] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1999.

[3] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'88*, 1988.

[4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[5] Philip J. Hayes and Steven P. Weinstein. Construe/tis: A system for content-based indexing of a database of news stories. In *Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence*, pages 49–64. AAAI Press, 1991.

[6] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers, 1998.

[7] Meenakshi Narayanaswamy, K. E. Ravikumar, and K. Vijay-Shanker. A biological named entity recognizer. In *Pacific Symposium on Biocomputing*, 2003.

[8] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.

[9] Zhou GuoDong and Su Jian. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of 2004 Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA'2004 shared task)*, pages 99–102, 2004.

[10] C. Blaschke, MA. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: protein-protein interaction. In *Proceedings of the AAAI conference on Intelligent Systems in Molecular Biology*, pages 60–7. AAAI, 1999.

[11] Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

[12] Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[13] Christian Blaschke and Alfonso Valencia. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, 17(2):14–20, 2002.

[14] Svetlana Novichkova, Sergei Egorov, and Nikolai Daraselia. MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, 19(13):1699–1706, 2003.

[15] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. In *Proceedings of the International Confernce on Intelligent Systems for Molecular Biology*, pages 574–82, 2001.

[16] Andrey Rzhetsky, Ivan Iossifov, Tomohiro Koike, Michael Krauthammer, Pauline Kra, Mitzi Morris, Hong Yu, Pablo Ariel Duboue, Wubin Weng, W. John Wilbur, Vasileios Hatzivassiloglou, and Carol Friedman. Geneways: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *J. of Biomedical Informatics*, 37(1):43–53, 2004.

[17] David P. A. Corney, Bernard F. Buxton, William B. Langdon, and David T. Jones. BioRAT: extracting biological information from full-length papers. *Bioinformatics*, 20(17):3206–3213, 2004.

[18] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[19] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Christos Andronis, Andreas Persidis, and Ourania Konstanti. Mining relations in the genia corpus. In *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, pages 61 – 68, 2004.

[20] Joshua M. Temkin and Mark R. Gilder. Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19(16):2046–2053, 2003.

[21] Gondy Leroy, Hsinchun Chen, Jesse D. Martinez, Shauna Eggers, Ryan R. Falsey, Kerri L. Kislin, Zan Huang, Jiexun Li, Jie Xu, Daniel M. McDonald, and Gavin Ng. Genescene: biomedical text and data mining. In *Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries*, pages 116–118. IEEE Computer Society, 2003.

[22] Jung-Hsien Chiang, Hsu-Chun Yu, and Huai-Jen Hsu. GIS: a biomedical text-mining system for gene information discovery. *Bioinformatics*, 20(1):120–121, 2004.

[23] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press, 1999.

[24] J. R. Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, 1990.

[25] J.S. Aitken. Learning information extraction rules: An inductive logic programming approach. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*. IOS Press, 2002.

[26] Kyu-Young Whang, Jongwoo Jeon, Kyuseok Shim, and Jaideep Srivastava. Advances in knowledge discovery and data mining, 7th pacific-asia conference, pakdd 2003, seoul, korea, april 30 - may 2, 2003, proceedings. In *PAKDD*, volume 2637 of *Lecture Notes in Computer Science*, pages 148 – 158. Springer, 2003.

[27] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. Comparative experiments on

learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 2003.

[28] Jing Ding, Daniel Berleant, Jun Xu, and Andy W. Fulmer. Extracting biochemical interactions from medline using a link grammar parser. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, page 467. IEEE Computer Society, 2003.

[29] Sampo Pyysalo, Filip Ginter, Tapio Pahikkala, Jeppe Koivula, Jorma Boberg, Jouni Jrvinen, and Tapio Salakoski. Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions inproceeding. In Adeline Nazarenko Nigel Collier, Patrick Ruch, editor, *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, number 0, pages 15–21, 2004.

[30] C. Fellbaum. Wordnet an electronic lexical database, 1998.

[31] Kurt W. Kohn. Molecular interaction map of the mammalian cell cycle control and dna repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, 1999.

[32] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[33] J.Castao, J.Zhang, and J.Pustejovsky. Anaphora resolution in biomedical literature. In *International Symposium on Reference Resolution*, 2002.

[34] D. D. Sleator and D. Temperley. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*, page ??, 1993.

[35] Dennis Grinberg, John Lafferty, and Daniel Sleator. A robust parsing algorithm for LINK grammars. Technical Report CMU-CS-TR-95-125, Pittsburgh, PA, 1995.

[36] P. Szolovits. Adding a medical lexicon to an english parser. pages 639–643, 2003.

[37] Ioannis Xenarios, Danny W. Rice, Lukasz Salwinski, Marisa K. Baron, Edward M. Marcotte, and David Eisenberg. DIP: the Database of Interacting Proteins. *Nucl. Acids Res.*, 28(1):289–291, 2000.

[38] Miguel Maroto, Ram Reshef, Andrea E. Mnsterberg, Susan Koester, Martyn Goulding, and Andrew B. Lassar. Ectopic pax-3 activates myod and myf-5 expression in embryonic mesoderm and neural tissue. *Cell*, 89(1):139–148, 1997.

APPENDIX A

USER MANUAL

The automated information extraction system has been developed using a combination of languages. The Pronoun resolution system has been written in prolog, and the link grammar parser is a C system. We have used Java to integrate all these module into a comprehensive system with an interactive user interface. We have used XSB prolog which provides a java API (interprolog) as a bridge to their prolog system. The link grammar parser required a wrapper class to be written that would process the output from the link grammar to an internal representation that can be used by the other java classes. The system has been developed for Windows. This appendix provides instructiuons to set up and use the system. The instructions are also duplicated in the INSTALL and README files provided with the software in the accompanying CD.

1. Installation

1. Unzip Extractor.zip to a location EXT_HOME in your hard disk. The source files and the class files are provided in the EXT_HOME/Extractor directory.

2. Unzip NLP.zip to C drive. The files should be in C:/NLP folder after unzipping.

Extractor.zip contains the source code, executables and related third party software needed for the information extraction module. The software needed by the system are

- XSB prolog - the prolog system from XSB
- Interprolog - the java api for XSB prolog
- Link grammar parser - the C system from Carnegir Mellon University

NLP.zip contains the Infogistics Natural Language Processor used by the extractor.

2. Setup

1. Include the library files in the CLASSPATH variable.
   (a) EXT_HOME/Extractor/interprolog201/interprolog.jar
   (b) C:/NLP/SDK/java/nlp.jar
2. Include the path to the XSB dll in the PATH variable

   path to XSB DLL: EXT_HOME/Extractor/xsb/config/x86-pc-windows/bin
3. Execute the Extractor
   (a) From Command Prompt. The command to execute is *java -Xmx500m IntExtractor*

   OR
   (b) Execute the batch file run.bat

3. Using the GUI

The User can extract interactions from :

1. A single abstract (as ASCII txt file).
   (a) From file menu select "Open Abstract", The Abstract contents are displayed in Text Area.
   (b) Click "Extract" button , the interactions extracted are displayed in the table. The interactions are written to a file named "Interactions.txt" located in folder named "Output".
2. Folder consisting of multiple abstracts (as ASCII txt files).

(a) From file menu select "Extract from Folder.

(b) Click "Extract" button , the interactions extracted are displayed in the table. The interactions are written to a file named "Interactions.txt" located in folder named "Output"

APPENDIX B

DATA AND CONTROL FLOW IN CODE

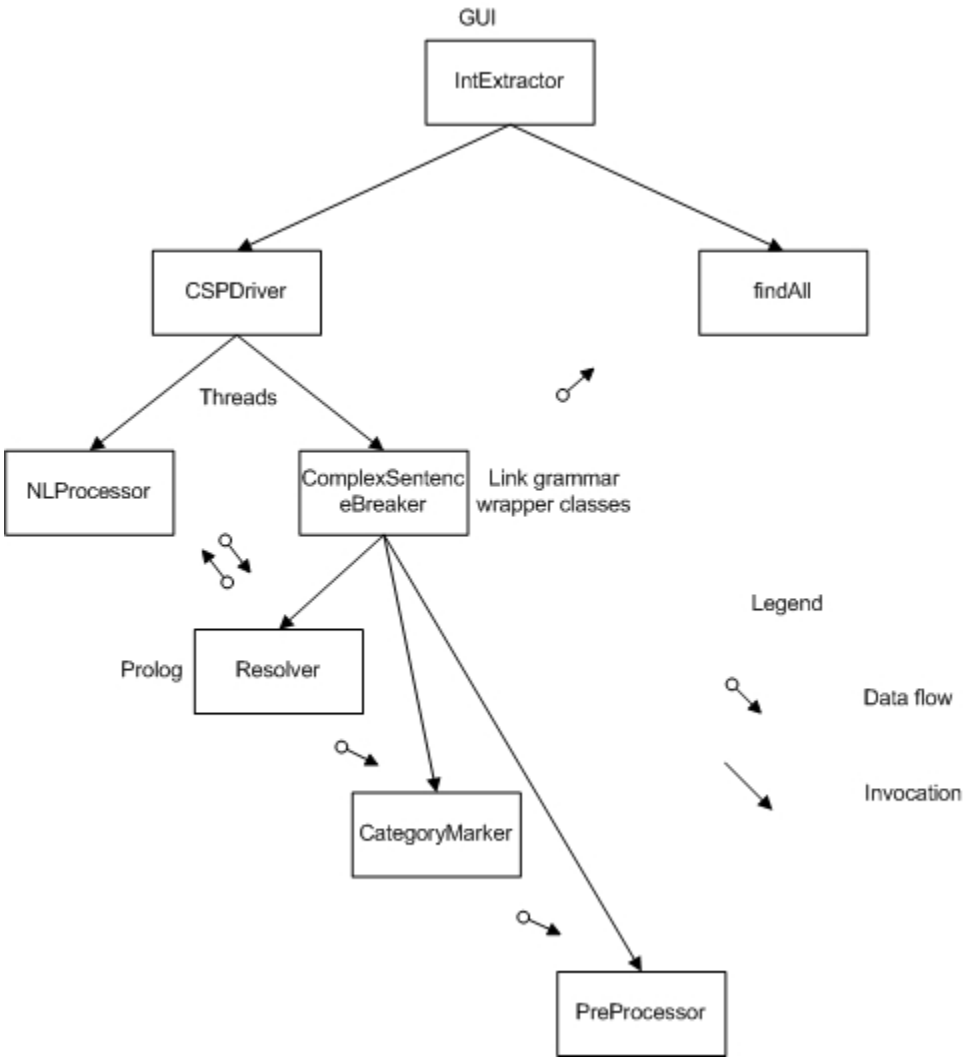

Figure 25. Control Flow in Code

APPENDIX C

ORGANIZATION OF FILES IN SOFTWARE

Directory	*Contents*
EXT_HOME	The Home directory where the Extractor.zip file has been extracted. This is where the Java classes reside in a default package. The link grammar parser executable is also present in this folder.
C:/NLP	The Folder containing Infogistics' Natural Language Processor.
EXT_HOME/prolog	The directory containing the prolog codes used for pronoun resolution.
EXT_HOME/data	The data used by the link grammar and the Entity tagging subsystem are placed in the data folder.
EXT_HOME/XSB	Contains the XSB prolog executables and DLLs used by the pronoun resolution subsystem.
EXT_HOME/interprolog201	Contains the files needed for the Java-Prolog bridge used to run the pronoun resolution subsystem in XSB prolog.
EXT_HOME/include	The files needed for the execution of the link grammar parser are placed here.
EXT_HOME/Output	The default output directory where the system stores the interactions extracted.
EXT_HOME/Example	Contains examples that can be texted on the system.
EXT_HOME/doc	Contains the javadoc for the java files.
BIORAT/abstracts	Contains the 229 abstracts used for testing obtained from the BioRAT people.

Table 2. Directory Structure in Software

The automated extraction system is distributed as single software integrating the Complex Sentence Processing and the Interaction Extraction modules into a single application with an interactive Java GUI.

The directory structure of the software is given in Table 2.