

CHAPTER 40

WEB-CONTENT MINING FOR LEARNING GENERIC RELATIONS AND THEIR ASSOCIATIONS FROM TEXTUAL BIOLOGICAL DATA

MUHAMMAD ABULAISH, PH.D.^{1,2} AND JAHIRUDDIN JAHIRUDDIN, M.C.A.²

¹Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

²Department of Computer Science, Jamia Millia Islamia (A Central University), New Delhi, India

1.1 INTRODUCTION

After sequencing of the human genome, the current bottleneck lies largely in the correct interpretation of the sequences [15]. In order to facilitate the understanding of the genome, a number of research efforts have been diverted in this direction and biologists are generating reams of biomedical literature. Since molecular biology has been a primary research area for more than last two decades, the number of text documents disseminating knowledge in this area has gone up manifolds and the explosion of literature makes it nearly impossible for a working biologist to keep up with developments in one field. The literature comprises accumulated knowledge in terms of the archival record of biological experiments, their methods, results and their interpretations. The sheer enormity of document collection in this domain necessitates the design of automated content analysis systems without which the assimilation of knowledge from this vast repository is becoming practically impossible [25]. Specialized search engines like PubMed have been designed to access information about these documents over the Web, but most of them uses simple pattern matching to answer user queries. Although, techniques such as simple pattern matching can highlight relevant text passages from large abstract collection, generating new insights to future research is far more complex.

PubMed [31] is a service of the *National Library of Medicine* [32] (NLM), USA that includes over 21 million citations for biomedical literatures from MEDLINE, life science journals, and online books. MEDLINE is the NLM's premier bibliographic database and forms the largest component of PubMed containing over 18 million citations dating back to the mid-1960's, covering all fields related to biomedicine [33]. In MEDLINE, the records are indexed with *Medical Subject Headings* [34] (MeSH) – a NLM's controlled vocabulary thesaurus containing sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity.

Until now, PubMed is the richest and most updated source of information about biological data despite its unstructured nature [8]. The result of a PubMed search is a list of citations (including authors, title, source, and often an abstract) to journal articles and an indication of free electronic full-text availability. In addition to this, PubMed provides other services including search filters for clinical queries, links to many other sites providing full-text articles and other related resources and citation matchers. Given a set of query terms, PubMed can identify research articles containing those terms quite efficiently [14].

In spite of these efforts, there is an increasing demand for intelligent *Information Retrieval* (IR) that requires analyzing the contextual relationship among query terms and judging the relevance of a document in the perspective of this relationship. For example, a simple query containing the string "*Alzheimer*" roughly translates to the requirement "*list all those documents that contain information about Alzheimer disease*" for which a simple pattern matching technique based on the occurrence of the query terms in a document is sufficient to decide whether it is relevant to the query or not. However, a more complex query in this domain can be expressed as "*metabolic ailments causes Alzheimer disease*", which translates to the requirement "*list all those documents that contain information about the metabolic ailments that causes Alzheimer disease.*" This is a much more complex query and requires contextual analysis of the query terms. As observed by Bernstein *et al.* [9], relating the entities in a query with a specific verb restricts the context of the concepts within text to a large extent. Hence, it is important that the relationships among the biomedical entities present in a text are also extracted and interpreted correctly.

Although, PubMed does not support contextual queries, it motivates the upsurge of interest in biomedical text mining to facilitate various degrees of automation in analyzing biological literature like *Named Entity Recognition* (NER), document classification, terminology extraction, relationship extraction and hypothesis generation [8]. Though, named-entity recognition, document classification and terminology extraction from biomedical text documents have gained reasonable success, reasoning about contents of a text document, however, needs more than identification of the entities present in it. Context of the entities in a document can be inferred from an analysis of the inter-entity relations present in the document.

Despite the fact that in addition to the development of many biomedical entity recognizers (e.g., ABNER [23], GENIA tagger [27], etc.) a number of approaches have been proposed to identify biological relations from texts [10, 20, 21, 22, 26], but most of them focus on mining a fixed set of biological relations occurring with a set of predefined tags. Thus, one of the pre-requisites for the success of these methods is the availability of tagged corpora in which biological entities are already marked. This is far from the reality – as most of the existing textual databases including PubMed do not perform annotations before storing scientific literatures. Moreover, each system is tuned to work with a pre-determined set of relations and does not address the problem of relation extraction in a generic way.

For example, the method of identification of interaction between genes and gene products cannot work for extraction of enzyme interactions from journal articles or for automatic extraction of protein interactions from scientific abstracts. Consideration of negation words like “not”, “neither”, etc. and morphological variants are also missing from most of the existing biomedical relation mining systems. In addition, to the best of our knowledge, none of the methods consider the extraction of *validatory entities*, while mining relational verbs and associated entities, whose presence or absence validates a particular biological interaction. For example, in the following PubMed sentence, “*regulates*” is identified as a relational verb relating the biological entities “*Rac1*” and “*transcription of the APP gene*” while “*primary hippocampal neurons*” can be identified as *validatory entity*, which restricts the scope of the regulation process mentioned in this sentence.

“... *Rac1 regulates transcription of the APP gene in primary hippocampal neurons* (PMID: 19267423).”

Since free texts are inherently unstructured or semi-structured in nature and difficult to interpret by computer programs, there has been increasing interest in recent past in apply text mining techniques to facilitate users to quickly perceive knowledge from the Web [2]. In contrast to existing text document processing techniques, which generally converts text documents into term vectors or bags-of-words, text mining process involves two subtasks – *text refining* and *knowledge distillation* [1]. The text refining task focuses on transforming free text into an intermediate machine-processable representation, whereas knowledge distillation analyzes the intermediate representation to deduce patterns or knowledge from it. In line with this approach of text mining process, in this chapter, we present the design of a web-content mining system that translates biological text documents into an intermediate representations (conceptual graph) using their syntax trees generated by the parser, which is then analyzed during knowledge distillation phase to identify information components comprising relational verbs and related constituents. The information components are thereafter analyzed to identifying feasible generic biological relations and their associations. Different categories of variants of a relational verb are recognized by our system. The first category comprises morphological variant of the root verb, which is essentially modification of the root verb itself. In English language the word *morphology* is usually categorized into *inflectional* and *derivational* morphology. Inflectional morphology studies the transformation of words for which the root form only changes, keeping the syntactic constraints invariable. For example, the root verb *activate*, has three inflectional verb forms – *activates*, *activated* and *activating*. Derivational morphology on the other hand deals with the transformation of the stem of a word to generate other words that retain the same concept, but may have different syntactic roles. Thus, *activate* and *activation* refer to the concept of “*making active*”, but one is a verb and the other one a noun. Similarly, *inactivate*, *transactivate*, *deactivate*, etc. are derived morphological variants created with addition of prefixes.

In the context of biological relations, we also observe that the occurrence of a verb in conjunction with a preposition very often changes the nature of the relation. For example, the focus of the relation “*activates*” may be quite different from the relation “*activates in*”, in which the verb “*activates*” is followed by the preposition “*in*”. Thus our system also considers a third category of biological relations, which are combinations of root verbs or their morphological variants, and prepositions that follow these. Typical examples of biological relations identified in this category include “*activated in*”, “*binds to*”, “*stimulated with*”, etc. This category of relations can take care of special biological

interactions involving substances and sources or localizations. Besides mining relational verbs with accompanying prepositions and associated entities, the entities associated with object entity through conjunctive prepositions are also extracted and termed as *validatory entities*, whose presence or absence validate a particular biological interaction.

Rest of this chapter is structured as follows. Starting with a brief review of the existing state-of-the-art in biological relation mining in section 1.2, we provide the functional detail of the proposed web-content mining system in section 1.3. The experimental setup and evaluation results are presented in section 1.4. Section 1.5 presents the uniqueness of the proposed relation mining system over existing ones. Finally, we conclude the chapter in section 1.6 with future directions of work.

1.2 STATE-OF-THE-ART IN BIOLOGICAL RELATION MINING

In this section, we present a brief review of the existing state-of-the-art in biological relation mining. Although, in addition to the development of biological entity recognition systems, a number of research efforts have been directed towards identifying associations between biological entities, most of the researchers have focused on extracting gene-gene, protein-protein, and gene-protein relations. Consequently, a number of relation mining techniques based on co-occurrence based approach, linguistic-based approach, or mixed-mode approach have been proposed by the researchers.

In co-occurrence based approach, relations between biological entities are inferred based on the assumption that two entities in the same sentence or abstract are related. Although, this approach is very simple to implement and computationally efficient, it provides high recall at the cost of poor precision and negation in sentences is usually ignored. Jenssen *et al.* [16] collected a set of almost 14,000 gene names from publicly available databases and used them to search MEDLINE abstracts. Two genes were assumed to be linked if they appeared in the same abstract; the relation received a higher weight if the gene pair appeared in multiple abstracts. The biological entity pairs occurring more than four times were assigned a high weight, and it was reported that 71% of such gene pairs were indeed related. However, the primary focus of the work is to extract related gene pairs rather than studying the nature of the relations. Albert *et al.* [6] used dictionaries of protein and interaction terms to retrieve protein-protein interaction from MEDLINE documents. They used tri-occurrences of two proteins and one interaction within a sentence for this purpose. This tri-occurrence extraction method enhances the recall at the cost of the precision. They reported overall precision of 22% only. Wren and Garner [29] identified related biological objects like genes, phenotypes, chemicals etc. by analyzing the cohesiveness and the specificity of the graph structure created by the co-occurrences of the objects within same MEDLINE records. Mukherjea and Sahay [19] presented a co-occurrence based technique to automatically discover biomedical relations from the World Wide Web. They first used the web search engine with manually-crafted lexicon-syntactic patterns to retrieve relevant information. Then, they used the extracted information to classify biomedical terms and discover relationships between biomedical entities.

In contrast to co-occurrence based approach, which does not exploit the linguistic features of text, linguistic-based approach usually applies parsing techniques to locate a set of handpicked *verbs* or *nouns*. Rules are specifically developed to extract the surrounding words of the pre-defined terms and to format them as relations. As with the co-occurrence

based approach, negation in sentences is usually ignored. Sekimizu *et al.* [22] collected the most frequently occurring verbs in a collection of abstracts and developed partial and shallow parsing techniques to find the verb's *subject* and *object*. The estimated precision of inferring relations is about 71%. Thomas *et al.* [26] modified a pre-existing parser based on cascaded finite state machines to fill templates with information on protein interactions for three verbs - *interact with*, *associate with*, and *bind to*. They calculated precision and recall in four different manners for three samples of abstracts. The precision values ranged from 60% to 81% and that the recall values from 24% to 63%. The PASTA system [13] is a more comprehensive system that extracts relations between proteins, species, and residues. It uses type and *Parts-Of-Speech* (POS) tagging along with manually created templates and lexicons assembled from biological databases to extract relationships between amino acid residues and their functions within a protein. This work reports precision of 82% and a recall value of 84% for recognition and classification of the terms, and 65% precision and 68% recall for completion of templates. Ono *et al.* [20] reported a method for extraction of protein-protein interactions based on a combination of syntactic patterns. They employed a dictionary look-up approach to identify proteins in text documents. Sentences that contain at least two proteins were selected and parsed with POS matching rules. The rules were triggered by a set of keywords that are frequently used to name protein interactions (e.g., *associate*, *bind*, etc.). Rinaldi *et al.* [21] proposed an approach towards automatic extraction of a pre-defined set of seven relations in the domain of molecular biology, based on a complete syntactic analysis of an existing corpus. They extracted relevant relations from a domain corpus based on full parsing of the documents and a set of rules that map syntactic structures into the relevant relations. Friedman *et al.* [11] developed a natural language processing system, GENIES, for the extraction of molecular pathways from journal articles. GENIES identifies a predefined set of verbs using templates for each one of these, which are encoded as a set of rules. This work reports a precision of 96% for identifying relations between biological molecules from full-text articles. Wattarujeekrit *et al.* [28] proposed a system, PASBio, to extract relation between verbs and its arguments by using *Predicate Argument Structure* (PAS). PASBio is specifically designed for annotating molecular events and defining core arguments that are important for completing the meaning of an event. Presently, PASBio contains the analyzed PAS of over 30 verbs. In [12], the authors proposed a system, Re1Ex, to extract relations between genes and proteins. For relation extraction, the text documents are first converted into dependency parse tree using Stanford lexicalized parser. Thereafter, rules are applied to identify candidate relations from parse trees. Both, precision and recall of the proposed system calculated over 1 million MEDLINE abstracts are reported as 80%. Xu *et al.* [30] proposed a method to extract relationship between gene and disease from literature. They used several strategies to filter out the sentences which do not contain relationship, then extracted the relationships between gene and disease by using the pattern of entities and relationship phrases. They reported the precision, recall and *F*-score values for their system as 84.6%, 77.5% and 80.9%, respectively.

Mixed-mode approach exploits both co-occurrence and linguistic features to identify relations between biological entities. Ciaramita *et al.* [10] reported an unsupervised learning mechanism for extracting semantic relations between molecular biology concepts from tagged MEDLINE abstracts. For each sentence containing two biological entities, a dependency graph highlighting the dependency between the entities is generated based on linguistic analysis. A relation between two entities is extracted as the shortest path between the pair following the dependency relations. The major emphasis of this work

is to determine the role of a concept in a significant relation and enhance biological ontology to include these roles and relations. Sentences containing complex embedded conjunctions/disjunctions or more than 100 words were not used for relation extraction. In the presence of nested tags, the system considers only the innermost tags. Miwa *et al.* [18] proposed a method to combine kernels based on several syntactic parsers for extracting protein-protein interactions from a given sentence. Their method used *Support Vector Machine* (SVM) and reported that their method achieve better results than other state-of-the-art *Protein-Protein-Interaction* (PPI) systems. Abulaish and Dey proposed an ontology-based *Biological Information Extraction and Query Answering* (BIEQA) System which extracts biological relations from MEDLINE abstracts using a series of natural language process techniques and co-occurrence based analysis from tagged documents [5]. Each mined relation is associated to a fuzzy membership value, which is proportional to its frequency of occurrence in the corpus and is termed a fuzzy biological relation. The fuzzy biological relations along with other relevant information components like biological entities occurring within a relation are stored in database which is integrated with a query-processing module. The query processing module has an interface, which guides users to formulate biological queries at different levels of specificity.

It can be observed that most of the systems have been developed to extract a pre-determined set of relations. The relation set is manually chosen to include a set of frequently occurring relations. Each system is tuned to work with a pre-determined set of relations and does not address the problem of relation extraction in a generic way. For example, the method of identification of interaction between genes and gene products cannot work for extraction of enzyme interactions from journal articles or for automatic extraction of protein interactions from scientific abstracts. Although, the methods proposed by Ciaramita *et al.* in [10] and Abulaish and Dey in [5] consider generic biomedical relation extraction, but both of them requires annotated text documents in which biomedical entities are already marked.

1.3 PROPOSED BIOLOGICAL RELATION MINING SYSTEM

In this section, we present the design and functional detail of the proposed biological relation mining system, which facilitates knowledge curation and relation identification from textual biological data. Figure 1.1 presents the complete architecture of the proposed relation mining system, in which dotted arrows show data-flow, whereas solid arrows are used to represent the inter-dependence between the modules. The proposed system performs four major tasks to identify biological relations and their associations – *document crawling*, *document pre-processing and parsing*, *information components extraction*, and *feasible biological relations identification*. The functional details of these tasks are presented in the following sub-sections.

1.3.1 Document Crawling

The purpose of this module is to download PubMed documents and store them on local machine for further processing. The crawler is implemented as an interactive module in Java programming language, which uses PubMed API to fetch documents in XML format and store them after parsing into structured database on local machine. Biomedical documents stored in PubMed database are available in XML format in which tags are defined using

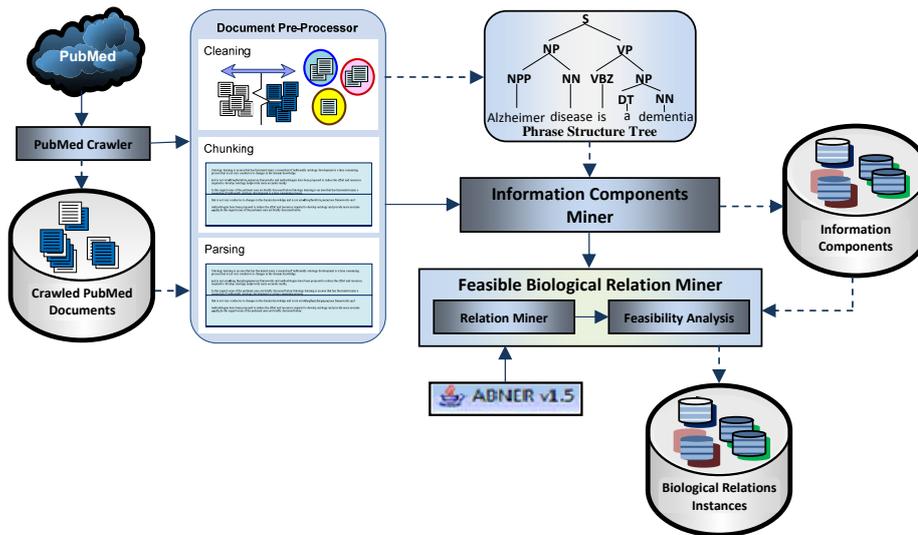


Figure 1.1 Architecture of the proposed biological relation mining system

Document Type Definition (DTD) file standardized by *World Wide Web Consortium* (W3C) [35]. The crawler uses DTD file definitions to create database schema to store fetched XML files from PubMed database into structured format. The fetched XML documents are parsed by crawler to identify different constituents like PMID, title, abstract, etc. to define the schema of the structured database. There are two types of APIs for parsing XML files – *tree-based Document Object Model* (DOM) and *event-based Simple API to XML* (SAX). Our crawler uses the SAX parser as DOM parser requires to read in and store the entire document in main memory prior to writing out any data and it is not possible for a large file that do not fit in the memory. However, the SAX parser receives data through a stream and recognizes the beginning and end of a document, element, or attribute in an event-driven manner. It writes out the data as it proceeds and there is no need to load entire file in the memory. After parsing XML files the *Java Database Connectivity* (JDBC) is used to store parsed data into a database.

1.3.2 Document Pre-processing and Parsing

The input to this module is the collection of text documents from which information components and biological relations are to be extracted. Initially, the input documents are cleaned through filtering meta-language tags and unwanted texts like author's names and affiliations, references, etc. A partial list of sample sentences to be filtered out during cleaning process is shown in table 1.1. The cleaned documents are tokenized into record-size chunks, boundaries of which are decided heuristically on the basis of the presence of various punctuation marks. Depending on the application, a record-size chunk may contain a sentence, a paragraph, or a complete document. Thereafter, the documents are parsed using a parser that assigns *Parts-Of-Speech* (POS) tags to every word in a sentence, where a tag reflects the syntactic category of the word [7]. POS analysis plays an important role in text information extraction since the syntactic category of a word determines its role in a

sentence to a large extent. The POS tags are useful to identify the grammatical structure of sentences like noun and verb phrases and their inter-relationships. For document parsing, we have used the Stanford parser [36], which is a statistical parser. The Stanford parser receives documents as input and works out the grammatical structure of sentences to convert them into equivalent phrase structure tree. A list of sample sentences and their corresponding phrase structure tree generated by Stanford parser is shown in table 1.2.

Table 1.1 A partial list of texts associated with PubMed abstracts that represent noise

PMID	Texts representing noise
20967920	Copyright ©2010 John Wiley & Sons, Ltd.
20967877	Cancer (Cancer Cytopathol) 2010. ©2010 American Cancer Society.
21221075	Laboratory Investigation advance online publication, 10 January 2011; doi:10.1038/labinvest.2010.199.
21219143	Please see http://www.annualreviews.org/catalog/pubdates.aspx for revised estimates.
21220675	Study Registration clinicalTrials.gov Identifier: NCT00106899.

1.3.3 Information Components Extraction

The concept of *information component* is introduced to capture and store the semantic structure of text into a structured format which can be used later on to apply association rule mining to identify the list of associated entities and their association strength with respect to a given corpus. Moreover, in line with the generalized associations mining technique proposed by Jiang *et al.* in [4], the extracted information components can also be used to mine generalized associations of generic biological relations identified by our proposed system. Since, the bag-of-words representation of a text document treat each representative term as an independent entity, the semantic relations depicting the conceptual roles are lost, i.e., terms lose their semantic relations and texts lose their original meanings [4]. For example, consider the following two sentences, whose bag-of-words representations (i.e., {*heart, disease, cause, depression*}) are same, but meaning is different. The first sentence (S1) represents the facts that “*heart disease*” causes “*depression*”, whereas the second sentence (S2) expresses an opposite meaning, i.e., “*depression*” causes “*heart disease*.”

Sentence-1 (S1): Heart disease causes depression

Sentence-2 (S2): Heart disease is caused by depression

Therefore, we have designed a set of rules to analyze text semantic structure (phrase structure tree generated by the parser) to identify *Noun Phrases* (NP) and *Verb Phrases* (VP), and their semantic relationship to generate conceptual graphs and then map them into information components. Since, the full conceptual graph standard is complex and could be computationally inefficient for knowledge distillation, we have used simplified conceptual graphs that are used in many existing researches [3, 4]. In conceptual graph, a node represents a NP or VP, whereas an edge represents a relation between them. In line with [4], three types of relations between the nodes (NP/VP) are identified to map the phrase structure tree of a sentence into a conceptual graph. $(i) < P, actor, Q >$, where P can be a VP and Q can be an NP or VP. In this relation, Q is an actor which performs

Table 1.2 Sample PubMed Sentences related to “*Alzheimer disease*” and their phrase structure tree representations generated by Stanford parser

PMID	PubMed sentence	Phrase structure tree representation
19295912	Transcriptome analysis of synaptoneuroosomes identifies neuroplasticity genes overexpressed in incipient Alzheimer’s disease.	(ROOT (S (NP (NP (JJ Transcriptome) (NN analysis)) (PP (IN of) (NP (NNS synaptoneuroosomes)))) (VP (VBZ identifies) (NP (NP (JJ neuroplasticity) (NNS genes)) (VP (VBN overexpressed) (PP (IN in) (NP (NP (JJ incipient) (NNP Alzheimer) (POS ’s)) (NN disease)))))) (. .)))
19295164	Recent studies suggest that bone marrow-derived macrophages can effectively reduce beta-amyloid (Abeta) deposition in brain.	(ROOT (S (NP (JJ Recent) (NNS studies)) (VP (VBP suggest) (SBAR (IN that) (S (NP (JJ bone) (JJ marrow-derived) (NNS macrophages)) (VP (MD can) (ADV (RB effectively)) (VP (VB reduce) (NP (NP (JJ beta-amyloid) (PRN (-LRB- -LRB-) (NP (NNP Abeta)) (-RRB- -RRB-)) (NN deposition)) (PP (IN in) (NP (NN brain)))))))) (. .)))
19275635	There is substantial and compelling evidence that aggregation and accumulation of amyloid beta protein (Abeta) plays a pivotal role in the development of Alzheimer’s disease (AD);	(ROOT (S (S (NP (EX There)) (VP (VBZ is) (NP (ADJP (JJ substantial) (CC and) (JJ compelling)) (NN evidence)) (SBAR (IN that) (S (NP (NP (NN aggregation) (CC and) (NN accumulation)) (PP (IN of) (NP (NP (JJ amyloid) (JJ beta) (NN protein)) (PRN (-LRB- -LRB-) (NP (NNP Abeta)) (-RRB- -RRB-)))))) (VP (VBZ plays) (NP (NP (DT a) (JJ pivotal) (NN role)) (PP (IN in) (NP (NP (DT the) (NN development)) (PP (IN of) (NP (NP (NNP Alzheimer) (POS ’s)) (NN disease)) (PRN (-LRB- -LRB-) (NNP AD) (-RRB- -RRB-)))))))))) (: :))
19263040	Memory deficits and neurochemical changes induced by C-reactive protein in rats: implication in Alzheimer’s disease.	(ROOT (NP (NP (NP (NN Memory) (NNS deficits) (CC and) (NN neurochemical) (NNS changes)) (VP (VBN induced) (PP (IN by) (NP (NP (JJ C-reactive) (NN protein)) (PP (IN in) (NP (NNS rats)))))) (: :)) (NP (NP (NN implication) (PP (IN in) (NP (NP (NNP Alzheimer) (POS ’s)) (NN disease)))) (. .)))
19293566	Conclusion: Although flanking SNP cover the whole gene transcript with strong linkage disequilibrium, our data show that the CST3 gene is not associated with AD risk in the Finnish population.	(ROOT (NP (NP (NNP Conclusion)) (: :)) (S (SBAR (IN Although) (S (VP (VBG flanking) (S (NP (NNP SNP)) (VP (VB cover) (NP (DT the) (JJ whole) (NN gene) (NN transcript)) (PP (IN with) (NP (JJ strong) (JJ linkage) (NN disequilibrium)))))) (, .) (NP (PRP\$ our) (NNS data)) (VP (VBP show) (SBAR (IN that) (S (NP (DT the) (NNP CST3) (NN gene)) (VP (VBZ is) (RB not) (VP (VBN associated) (PP (IN with) (NP (NNP AD) (NN risk)) (PP (IN in) (NP (DT the) (JJ Finnish) (NN population)))))))) (. .)))

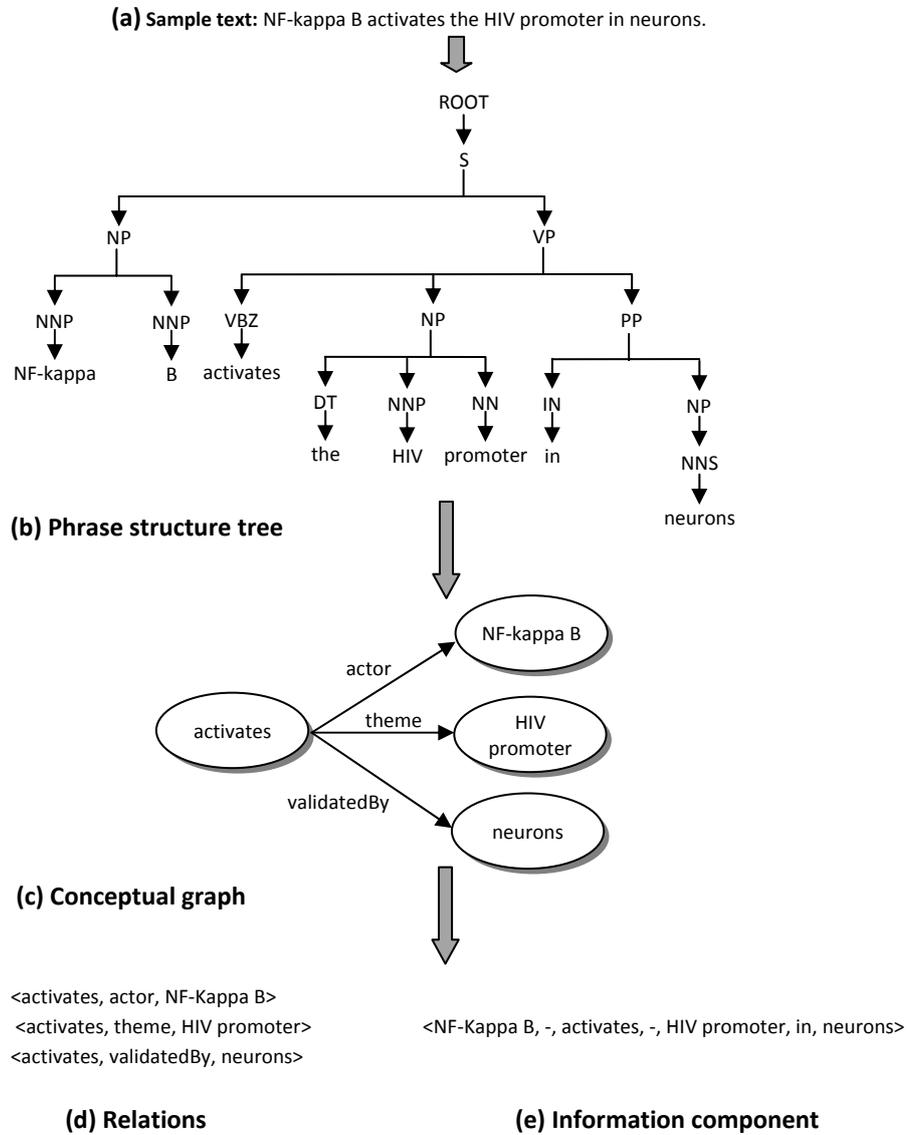


Figure 1.2 (a) A sample biological sentence, (b) phrase structure tree generated by the parser, (c) conceptual graph to model semantic structure of the text, (d) extracted relation instances from the conceptual graph, (e) generated information component

action P . For example, in figure 1.2 the relation $\langle \text{activates}, \text{actor}, \text{NF-KappaB} \rangle$ is an instance of this type of relation. (ii) $\langle P, \text{theme}, Q \rangle$, where P can be a VP and Q can be an NP or VP. In this relation, Q is a theme of the action P . For example, in figure 1.2 the relation $\langle \text{activates}, \text{theme}, \text{HIVpromoter} \rangle$ is an instance of this type of relation. (iii) $\langle P, \text{validatedBy}, Q \rangle$, where both P and Q can be an NP or VP. In this relation, P is validated by Q through a proposition. For example, in figure 1.2 the relation $\langle \text{activates}, \text{validatedBy}, \text{neurons} \rangle$ is an instance of this type of relation, which represents the fact that the activation is performed in “neurons.”

As discussed in [4], although a relation can be modeled directly using the template $\langle \text{subject}, \text{verb}, \text{object} \rangle$, but it fails to model a relation in which either *subject* or *object* is missing, specially in the case when a sentence is in passive form. Once the relations are identified from conceptual graph, they are clubbed together to create an instance of information component, which is defined in the following paragraph. Individual relations are also stored in a structured repository to mine relation associations. Besides, storing constituents from different relations extracted from a conceptual graph, the information component generation process also contains *adverbs* and *prepositions* to represent negations and state of biological interactions, respectively. Figure 1.2 presents a sample sentence, its phrase structure tree generated by the parser, conceptual graph, identified instances and information components.

Definition 1.1 (Information Component). An *Information Component* (IC) is a 7-tuple of the form $\langle E_i, A, V, P_v, E_j, P_c, E_k \rangle$ where, E_i , and E_j are noun phrases associated by V which is a relational verb; A is adverb; P_v is verbal-preposition associated with V ; E_k is validity phrase associated with E_j through conjunctive-preposition P_c .

Semantic tree analysis and information component extraction process is implemented as a rule-based system as shown in table 1.3. Dependencies output by the parser are analyzed to identify noun and verb phrases and their semantic relations. The algorithm 1.1, `informationComponentExtraction`, presents the implementation detail of the proposed rule-based system in a formal way. A partial list of information components extracted by this algorithm from PubMed sentences of table 1.2 is shown in table 1.4.

Table 1.3: Rules for analyzing phrase structure tree to identify information components

Rule no.	Rule statement
1.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_l(VP, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, \text{null}, V, \text{null}, E_j, \text{null}, \text{null} \rangle$
2.	$[C(R, E_i) \wedge C(R, VP) \wedge C(R, Adv) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_l(VP, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, Adv, V, \text{null}, E_j, \text{null}, \text{null} \rangle$
3.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_l(VP_2, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, \text{null}, V, \text{null}, E_j, \text{null}, \text{null} \rangle$
4.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_l(VP_2, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, Adv, V, \text{null}, E_j, \text{null}, \text{null} \rangle$
5.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, VP_3) \wedge C_l(VP_3, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, \text{null}, V, \text{null}, E_j, \text{null}, \text{null} \rangle$

6.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, Adv) \wedge C(VP_2, VP_3) \wedge L(VP_3, Adv) \wedge C_i(VP_3, V) \wedge S(V, E_j)] \Rightarrow \langle E_i, Adv, V, null, E_j, null, null \rangle$
7.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, null, V, p, E_j, null, null \rangle$
8.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, Adv, V, p, E_j, null, null \rangle$
9.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_i(VP_2, V) \wedge S(V, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, null, V, p, E_j, null, null \rangle$
10.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_i(VP_2, V) \wedge S(V, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, Adv, V, p, E_j, null, null \rangle$
11.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge C(ADVP, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, null, V, p, E_j, null, null \rangle$
12.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge C(ADVP, PP) \wedge C_i(PP, p) \wedge S(p, E_j)] \Rightarrow \langle E_i, Adv, V, p, E_j, null, null \rangle$
13.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$
14.	$[C(R, E_i) \wedge C(R, VP) \wedge C(R, Adv) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$
15.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$
16.	$[C(R, E_i) \wedge C(R, VP) \wedge C(R, Adv) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$
17.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_i(VP_2, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$
18.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_i(VP_2, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$
19.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_i(VP_2, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$
20.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_i(VP_2, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$
21.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, VP_3) \wedge C_i(VP_3, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$
22.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, Adv) \wedge C(VP_2, VP_3) \wedge L(VP_3, Adv) \wedge C_i(VP_3, V) \wedge S(V, E_j) \wedge S(V, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$
23.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, VP_3) \wedge C_i(VP_3, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, null, V, null, E_j, p, E_k \rangle$

24.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C(VP_2, Adv) \wedge C(VP_2, VP_3) \wedge L(VP_3, Adv) \wedge C_i(VP_3, V) \wedge S(V, NP) \wedge C(NP, E_j) \wedge C(NP, PP) \wedge L(PP, E_j) \wedge C_i(PP, p) \wedge S(p, E_k)] \Rightarrow \langle E_i, Adv, V, null, E_j, p, E_k \rangle$		
25.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, PP_1) \wedge S(V, PP_2) \wedge L(PP_2, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
26.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, PP_1) \wedge S(V, PP_2) \wedge L(PP_2, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
27.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
28.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
29.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_i(VP_2, V) \wedge S(V, PP_1) \wedge S(V, PP_2) \wedge L(PP_2, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
23.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_i(VP_2, V) \wedge S(V, PP_1) \wedge S(V, PP_2) \wedge L(PP_2, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
31.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, VP_2) \wedge C_i(VP_2, V) \wedge S(V, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
32.	$[C(R, E_i) \wedge C(R, VP_1) \wedge L(VP_1, E_i) \wedge C(VP_1, Adv) \wedge C(VP_1, VP_2) \wedge L(VP_2, Adv) \wedge C_i(VP_2, V) \wedge S(V, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
33.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge S(V, PP_2) \wedge L(PP_2, ADVP) \wedge C(ADVP, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
34.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge S(V, PP_2) \wedge L(PP_2, ADVP) \wedge C(ADVP, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
35.	$[C(R, E_i) \wedge C(R, VP) \wedge L(VP, E_i) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge C(ADVP, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, null, V, p_1, E_j, p_2, E_k \rangle$		
36.	$[C(R, E_i) \wedge C(R, Adv) \wedge C(R, VP) \wedge L(Adv, E_i) \wedge L(VP, Adv) \wedge C_i(VP, V) \wedge S(V, ADVP) \wedge C(ADVP, PP_1) \wedge C_i(PP_1, p_1) \wedge S(p_1, NP) \wedge C(NP, E_j) \wedge C(NP, PP_2) \wedge L(PP_2, E_j) \wedge C_i(PP_2, p_2) \wedge S(p_2, E_k)] \Rightarrow \langle E_i, Adv, V, p_1, E_j, p_2, E_k \rangle$		
Legend:	E_i, E_j, E_k : Entity appearing as noun phrase NP	$L(X, Y)$: Y is left to X	
R : Root of sub tree of the phrase structure tree	$C(X, Y)$: Y is child of X	$C_i(X, Y)$: Y is left most child of X	$S(X, Y)$: X and Y are sibling

Algorithm 1.1*informationComponentExtraction(T)***Input:** Phrase structure tree T , created through Stanford parser**Output:** A list of Information Components L_{IC} **Steps:**

```

1.  $L_{IC} \leftarrow \phi$ 
2. for each node  $N \in T$  do
3.   for each child  $\eta_i \in N$  do
4.      $IC \leftarrow \phi$ 
5.     if  $\eta_{i1} = NP$  AND  $\eta_{i2} = VP$  AND  $i1 < i2$  AND  $\alpha_0 \in \text{child}[\eta_{i2}] = V$  then
6.       if  $\alpha_j \in \text{child}[\eta_{i2}] = NP$  AND  $j \neq 0$  then
7.         if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p$ 
8.           AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then
9.              $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\alpha_j), p, E(\beta_{k2}) \rangle$  // Rule-13
10.            //  $E(x)$  represent the entity extracted from the subtree rooted at  $x$ .
11.          else if  $\beta_{k1} \in \text{child}[\alpha_{k1}] = NP$  AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = PP$  AND  $k1 < k2$ 
12.            AND  $\lambda_0 \in \text{child}[\beta_{k2}] = p$  AND  $\lambda_{k3} \in \text{child}[\beta_{k2}] = NP$  AND  $k3 \neq 0$  then
13.               $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\beta_{k1}), p, E(\lambda_{k3}) \rangle$  // Rule-15
14.            else
15.               $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\alpha_j), \text{null}, \text{null} \rangle$  // Rule-1
16.            end if
17.          else if  $\alpha_{j1} \in \text{child}[\eta_{i2}] = ADVP$  AND  $j1 \neq 0$  AND  $\beta_{j2} \in \text{child}[\alpha_{j1}] = PP$ 
18.            AND  $\lambda_0 \in \text{child}[\beta_{j2}] = p1$  AND  $\lambda_{j3} \in \text{child}[\beta_{j2}] = NP$  AND  $j3 \neq 0$  then
19.              if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p2$ 
20.                AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then
21.                   $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\lambda_{j3}), p2, E(\beta_{k2}) \rangle$  // Rule-33
22.                else if  $\gamma_{k1} \in \text{child}[\lambda_{j3}] = NP$  AND  $\gamma_{k2} \in \text{child}[\lambda_{j3}] = PP$  AND  $k1 < k2$ 
23.                  AND  $\theta_0 \in \text{child}[\gamma_{k2}] = p2$  AND  $\theta_{k3} \in \text{child}[\gamma_{k2}] = NP$  AND  $k3 \neq 0$  then
24.                     $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\gamma_{k1}), p2, E(\theta_{k3}) \rangle$  // Rule-35
25.                  else
26.                     $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\lambda_{j3}), \text{null}, \text{null} \rangle$  // Rule-11
27.                  end if
28.                else if  $\alpha_{j1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 \neq 0$  AND  $\beta_0 \in \text{child}[\alpha_{j1}] = p1$ 
29.                  AND  $\beta_{j2} \in \text{child}[\alpha_{j1}] = NP$  AND  $j2 \neq 0$  then
30.                    if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p2$ 
31.                      AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then
32.                         $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\beta_{j2}), p2, E(\beta_{k2}) \rangle$  // Rule-25
33.                      else if  $\lambda_{k1} \in \text{child}[\beta_{j2}] = NP$  AND  $\lambda_{k2} \in \text{child}[\beta_{j2}] = PP$  AND  $k1 < k2$ 
34.                        AND  $\gamma_0 \in \text{child}[\lambda_{k2}] = p2$  AND  $\gamma_{k3} \in \text{child}[\lambda_{k2}] = NP$  AND  $k3 \neq 0$  then
35.                           $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\lambda_{k1}), p2, E(\gamma_{k3}) \rangle$  // Rule-27
36.                        else
37.                           $IC = \langle E(\eta_{i1}), \text{null}, V, p1, E(\beta_{j2}), \text{null}, \text{null} \rangle$  // Rule-7
38.                        end if
39.                      end if
40.                    else if  $\eta_{i1} = NP$  AND  $\eta_{i2} = VP$  AND  $i1 < i2$  AND  $\alpha_{i3} \in \text{child}[\eta_{i2}] = VP$ 
41.                      AND  $\beta_0 \in \text{child}[\alpha_{i3}] = V$  then
42.                        if  $\beta_j \in \text{child}[\alpha_{i3}] = NP$  AND  $j \neq 0$  then
43.                          if  $\beta_{k1} \in \text{child}[\alpha_{i3}] = PP$  AND  $j < k1$  AND  $\lambda_0 \in \text{child}[\beta_{k1}] = p$ 
44.                            AND  $\lambda_{k2} \in \text{child}[\beta_{k1}] = NP$  AND  $k2 \neq 0$  then
45.                               $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\beta_j), p, E(\lambda_{k2}) \rangle$  // Rule-17
46.                            else if  $\lambda_{k1} \in \text{child}[\beta_j] = NP$  AND  $\lambda_{k2} \in \text{child}[\beta_j] = PP$  AND  $k1 < k2$ 
47.                              AND  $\gamma_0 \in \text{child}[\lambda_{k2}] = p$  AND  $\gamma_{k3} \in \text{child}[\lambda_{k2}] = NP$  AND  $k3 \neq 0$  then
48.                                 $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\lambda_{k1}), p, E(\gamma_{k3}) \rangle$  // Rule-19
49.                              else
50.                                 $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\beta_j), \text{null}, \text{null} \rangle$  // Rule-3
51.                              end if
52.                            else if  $\beta_{j1} \in \text{child}[\alpha_{i3}] = PP$  AND  $j1 \neq 0$  AND  $\lambda_0 \in \text{child}[\beta_{j1}] = p1$ 
53.                              AND  $\lambda_{j2} \in \text{child}[\beta_{j1}] = NP$  AND  $j2 \neq 0$  then
54.                                if  $\beta_{k1} \in \text{child}[\alpha_{i3}] = PP$  AND  $j1 < k1$  AND  $\lambda_0 \in \text{child}[\beta_{k1}] = p2$ 

```

```

42.     AND  $\lambda_{k2} \in \text{child}[\beta_{k1}] = NP$  AND  $k2 \neq 0$  then
43.          $IC = \langle E(\eta_{i1}), \text{null}, V, p_1, E(\lambda_{j2}), p_2, E(\lambda_{k2}) \rangle$  // Rule-29
44.     else if  $\gamma_{k1} \in \text{child}[\lambda_{j2}] = NP$  AND  $\gamma_{k2} \in \text{child}[\lambda_{j2}] = PP$  AND  $k1 < k2$ 
45.         AND  $\theta_0 \in \text{child}[\gamma_{k2}] = p_2$  AND  $\theta_{k3} \in \text{child}[\gamma_{k2}] = NP$  AND  $k3 \neq 0$  then // Rule-31
46.          $IC = \langle E(\eta_{i1}), \text{null}, V, p_1, E(\gamma_{k1}), p_2, E(\theta_{k3}) \rangle$ 
47.     else
48.          $IC = \langle E(\eta_{i1}), \text{null}, V, p_1, E(\lambda_{j2}), \text{null}, \text{null} \rangle$  // Rule-9
49.     end if
50. else if  $\eta_{i1} = NP$  AND  $\eta_{i2} = VP$  AND  $i1 < i2$  AND  $\alpha_{i3} \in \text{child}[\eta_{i2}] = VP$ 
51. AND  $\beta_{i4} \in \text{child}[\alpha_{i3}] = VP$  AND  $\lambda_0 \in \text{child}[\beta_{i4}] = V$ 
52. AND  $\lambda_{i5} \in \text{child}[\beta_{i4}] = NP$  AND  $i5 \neq 0$  then
53.     if  $\lambda_{j1} \in \text{child}[\beta_{i4}] = PP$  AND  $i5 < j1$  AND  $\gamma_0 \in \text{child}[\lambda_{j1}] = p$ 
54.     AND  $\gamma_{j2} \in \text{child}[\lambda_{j1}] = NP$  AND  $j2 \neq 0$  then
55.          $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\lambda_{i5}), p, E(\gamma_{j2}) \rangle$  // Rule-21
56.     else if  $\gamma_{j1} \in \text{child}[\lambda_{i5}] = NP$  AND  $\gamma_{j2} \in \text{child}[\lambda_{i5}] = PP$  AND  $j1 < j2$ 
57.     AND  $\theta_0 \in \text{child}[\gamma_{j2}] = p$  AND  $\theta_{j3} \in \text{child}[\gamma_{j2}] = NP$  AND  $j3 \neq 0$  then // Rule-23
58.          $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\gamma_{j1}), p, E(\theta_{j3}) \rangle$ 
59.     else
60.          $IC = \langle E(\eta_{i1}), \text{null}, V, \text{null}, E(\lambda_{i5}), \text{null}, \text{null} \rangle$  // Rule-5
61.     end if
62. else if  $\eta_{i1} = NP$  AND  $\eta_{i2} = VP$  AND  $\eta_{i3} = Adv$  AND  $i1 < i2 < i3$ 
63. AND  $\alpha_0 \in \text{child}[\eta_{i2}] = V$  then
64.     if  $\alpha_j \in \text{child}[\eta_{i2}] = NP$  AND  $j \neq 0$  then
65.         if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p$ 
66.         AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then // Rule-14
67.              $IC = \langle E(\eta_{i1}), Adv, V, \text{null}, E(\alpha_j), p, E(\beta_{k2}) \rangle$ 
68.         else if  $\beta_{k1} \in \text{child}[\alpha_{k1}] = NP$  AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = PP$  AND  $k1 < k2$ 
69.         AND  $\lambda_0 \in \text{child}[\beta_{k2}] = p$  AND  $\lambda_{k3} \in \text{child}[\beta_{k2}] = NP$  AND  $k3 \neq 0$  then // Rule-16
70.              $IC = \langle E(\eta_{i1}), Adv, V, \text{null}, E(\beta_{k1}), p, E(\lambda_{k3}) \rangle$ 
71.         else
72.              $IC = \langle E(\eta_{i1}), Adv, V, \text{null}, E(\alpha_j), \text{null}, \text{null} \rangle$  // Rule-2
73.         end if
74.     else if  $\alpha_{j1} \in \text{child}[\eta_{i2}] = ADVP$  AND  $j1 \neq 0$  AND  $\beta_{j2} \in \text{child}[\alpha_{j1}] = PP$ 
75.     AND  $\lambda_0 \in \text{child}[\beta_{j2}] = p_1$  AND  $\lambda_{j3} \in \text{child}[\beta_{j2}] = NP$  AND  $j3 \neq 0$  then
76.     if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p_2$ 
77.     AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then // Rule-34
78.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\lambda_{j3}), p_2, E(\beta_{k2}) \rangle$ 
79.     else if  $\gamma_{k1} \in \text{child}[\lambda_{j3}] = NP$  AND  $\gamma_{k2} \in \text{child}[\lambda_{j3}] = PP$  AND  $k1 < k2$ 
80.     AND  $\theta_0 \in \text{child}[\gamma_{k2}] = p_2$  AND  $\theta_{k3} \in \text{child}[\gamma_{k2}] = NP$  AND  $k3 \neq 0$  then // Rule-36
81.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\gamma_{k1}), p_2, E(\theta_{k3}) \rangle$ 
82.     else
83.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\lambda_{j3}), \text{null}, \text{null} \rangle$  // Rule-12
84.     end if
85.     else if  $\alpha_{j1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 \neq 0$  AND  $\beta_0 \in \text{child}[\alpha_{j1}] = p_1$ 
86.     AND  $\beta_{j2} \in \text{child}[\alpha_{j1}] = NP$  AND  $j2 \neq 0$  then
87.     if  $\alpha_{k1} \in \text{child}[\eta_{i2}] = PP$  AND  $j1 < k1$  AND  $\beta_0 \in \text{child}[\alpha_{k1}] = p_2$ 
88.     AND  $\beta_{k2} \in \text{child}[\alpha_{k1}] = NP$  AND  $k2 \neq 0$  then // Rule-26
89.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\beta_{j2}), p_2, E(\beta_{k2}) \rangle$ 
90.     else if  $\lambda_{k1} \in \text{child}[\beta_{j2}] = NP$  AND  $\lambda_{k2} \in \text{child}[\beta_{j2}] = PP$  AND  $k1 < k2$ 
91.     AND  $\gamma_0 \in \text{child}[\lambda_{k2}] = p_2$  AND  $\gamma_{k3} \in \text{child}[\lambda_{k2}] = NP$  AND  $k3 \neq 0$  then // Rule-28
92.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\lambda_{k1}), p_2, E(\gamma_{k3}) \rangle$ 
93.     else
94.          $IC = \langle E(\eta_{i1}), Adv, V, p_1, E(\beta_{j2}), \text{null}, \text{null} \rangle$  // Rule-8
95.     end if
96. end if
97. else if  $\eta_{i1} = NP$  AND  $\eta_{i2} = VP$  AND  $i1 < i2$  AND  $\alpha_{i3} \in \text{child}[\eta_{i2}] = VP$ 
98. AND  $\alpha_{i4} \in \text{child}[\eta_{i2}] = Adv$  AND  $i4 < i3$  AND  $\beta_0 \in \text{child}[\alpha_{i3}] = V$  then
99.     if  $\beta_j \in \text{child}[\alpha_{i3}] = NP$  AND  $j \neq 0$  then
100.        if  $\beta_{k1} \in \text{child}[\alpha_{i3}] = PP$  AND  $j < k1$  AND  $\lambda_0 \in \text{child}[\beta_{k1}] = p$ 
101.        AND  $\lambda_{k2} \in \text{child}[\beta_{k1}] = NP$  AND  $k2 \neq 0$  then

```

```

86.      IC = ⟨E(ηi1), Adv, V, null, E(βj), p, E(λk2)⟩ // Rule-18
87.      else if λk1 ∈ child[βj] = NP AND λk2 ∈ child[βj] = PP AND k1 < k2
      AND γ0 ∈ child[λk2] = p AND γk3 ∈ child[λk2] = NP AND k3 ≠ 0 then
88.      IC = ⟨E(ηi1), Adv, V, null, E(λk1), p, E(γk3)⟩ // Rule-20
89.      else
90.      IC = ⟨E(ηi1), Adv, V, null, E(βj), null, null⟩ // Rule-4
91.      end if
92.      else if βj1 ∈ child[αi3] = PP AND j1 ≠ 0 AND λ0 ∈ child[βj1] = p1
      AND λj2 ∈ child[βj1] = NP AND j2 ≠ 0 then
93.      if βk1 ∈ child[αi3] = PP AND j1 < k1 AND λ0 ∈ child[βk1] = p2
      AND λk2 ∈ child[βk1] = NP AND k2 ≠ 0 then
94.      IC = ⟨E(ηi1), Adv, V, p1, E(λj2), p2, E(λk2)⟩ // Rule-30
95.      else if γk1 ∈ child[λj2] = NP AND γk2 ∈ child[λj2] = PP AND k1 < k2
      AND θ0 ∈ child[γk2] = p2 AND θk3 ∈ child[γk2] = NP AND k3 ≠ 0 then
96.      IC = ⟨E(ηi1), Adv, V, p1, E(γk1), p2, E(θk3)⟩ // Rule-32
97.      else
98.      IC = ⟨E(ηi1), Adv, V, p1, E(λj2), null, null⟩ // Rule-10
99.      end if
100.     end if
101.     else if ηi1 = NP AND ηi2 = VP AND i1 < i2 AND αi3 ∈ child[ηi2] = VP
      AND βi4 ∈ child[αi3] = VP AND βi6 ∈ child[αi3] = Adv AND i6 < i4
      AND λ0 ∈ child[βi4] = V AND λi5 ∈ child[βi4] = NP AND i5 ≠ 0 then
102.     if λj1 ∈ child[βi4] = PP AND i5 < j1 AND γ0 ∈ child[λj1] = p
      AND γj2 ∈ child[λj1] = NP AND j2 ≠ 0 then
103.     IC = ⟨E(ηi1), Adv, V, null, E(λi5), p, E(γj2)⟩ // Rule-22
104.     else if γj1 ∈ child[λi5] = NP AND γj2 ∈ child[λi5] = PP AND j1 < j2
      AND θ0 ∈ child[γj2] = p AND θj3 ∈ child[γj2] = NP AND j3 ≠ 0 then
105.     IC = ⟨E(ηi1), Adv, V, null, E(γj1), p, E(θj3)⟩ // Rule-24
106.     else
107.     IC = ⟨E(ηi1), Adv, V, null, E(λi5), null, null⟩ // Rule-6
108.     end if
109.     end if
110.     if IC ≠ φ then
111.     LIC ← LIC ∪ IC
112.     end if
113.     end for
114.     end for
115.     Return LIC

```

Table 1.4 A partial list of information components extracted from the sample sentences related to “Alzheimer disease” of table 1.2

Left entity	Adv	Relational verb	Verbal prep.	Right entity	Conj. prep.	Validatory entity	PMID
Transcriptome analysis of synaptoneuroosomes	—	identifies	—	neuroplasticity genes overexpressed in incipient Alzheimer’s disease	—	—	19295912
neuroplasticity genes	—	overexpressed	in	incipient Alzheimer’s disease	—	—	19295912
bone marrow-derived macrophages	—	reduce	—	beta-amyloid (Abeta) deposition	in	brain	19295164
aggregation and accumulation of amyloid beta protein (Abeta)	—	plays	—	a pivotal role	in	the development of Alzheimer’s disease (AD)	19275635
Memory deficits and neurochemical changes	—	induced	by	C-reactive protein	in	rats	19263040
the CST3 gene	not	associated	with	AD risk	in	the Finnish population	19263040

1.3.4 Feasible Biological Relations Identification

A biomedical relation is usually manifested in a document as a relational verb associating two or more biological entities. The biological actors associated to a relation can be inferred from the entities located in the proximity of the relational verb. At present, we have considered only binary relations. Since relation instances specified at entity-levels are rare, while applying mining techniques on them the support count of many itemsets would be very low. Therefore, the biological entities appearing in information components are marked with a biological entity recognizer that helps in identifying valid biological relations and their associations. For this purpose, our system is integrated with a biological named entity recognizer, ABNER (v1.5) [23], which is a molecular biology text analysis tool. ABNER employs statistical machine learning using linear-chain *Conditional Random Fields* (CRFs) with a variety of orthographic and contextual features and it is trained on both the NLPBA and *BioCreative* corpora. In order to compile biological relations from information components, we consider only those information components in which either left entity or right entity field has at least one biomedical entity. In this way, a large number of irrelevant verbs are eliminated from being considered as biological relations. Further irrelevant relational verbs are eliminated by applying the following definition of the *feasible biological relation*:

Definition 1.2 (*Feasible Biological Relation*). A relational verb V is said to be a *feasible biological relation* with respect to a given corpus if the support count of V in proximity of biological entities is greater than a threshold value θ .

The feasibility analysis helps in eliminating a number of relational verbs which may have chance occurrence in biological domain. These verbs usually represent author biases and their elimination reduces the overall computational load. For example, the verbs “*worked with*”, “*experimented with*”, “*found*”, etc. may occur in a few technical articles, but not frequent enough to be considered as a significant term for biological domain. Since, our aim is not just to identify possible relational verbs, but to identify feasible biological relations, we engage in statistical analysis to identify feasible biological relations. To consolidate the final list of feasible relations we take care of two things. Firstly, since various forms of the same verb represent a basic biological relation in different forms, the feasible collection is extracted by considering only the unique root forms after analyzing the complete list of information components. The root verb having support count greater than or equal to a threshold value is retained as root biological relations. Thereafter, information components are again analyzed to identify the morphological variants of the retained root verbs using partial pattern matching technique. The Algorithm 1.2, `biomedicalRelationExtraction`, defines this process formally. A partial list of feasible biological relations and their morphological variants extracted from a corpus of 500 PubMed abstracts related to *Alzheimer disease* is shown in table 1.5.

Algorithm 1.2

biomedicalRelationExtraction(L_{IC})

Input: L_{IC} - A list of information components

Output: A set R of feasible biological relations and their morphological variants

Steps:

```

1.  $L_V \leftarrow \phi, L_{UV} \leftarrow \phi, L_{RV} \leftarrow \phi$ 
2. for all  $IC \in L_{IC}$  do
3.   if  $E_i \in IC.leftEntity$  OR  $E_i \in IC.rightEntity$  then
4.      $L_V \leftarrow L_V \cup IC.verb + IC.preposition$  //  $E_i$  is biological entity identified by ABNER
5.   end if
6. end for
7.  $L_{UV} \leftarrow UNIQUE(L_V)$  // create a list of unique verbs
8. Filter out verbs from  $L_{UV}$  with a prefix as  $\xi$ , where  $\xi \in \{\text{cross-}, \text{extra-}, \text{hydro-}, \text{micro-}, \text{milli-}, \text{multi-}, \text{photo-}, \text{super-}, \text{anti-}, \text{down-}, \text{half-}, \text{hypo-}, \text{mono-}, \text{omni-}, \text{over-}, \text{poly-}, \text{self-}, \text{semi-}, \text{tele-}, \text{dis-}, \text{epi-}, \text{mis-}, \text{non-}, \text{pre-}, \text{sub-}, \text{de-}, \text{di-}, \text{il-}, \text{im-}, \text{ir-}, \text{un-}, \text{up-}\}$ 
9. Filter out verbs from  $L_{UV}$  with a suffix as  $\lambda$ , where  $\lambda \in \{-able, -tion, -ness, -less, -ment, -ally, -ity, -ism, -ous, -ing, -er, -or, -al, -ly, -ed, -es, -ts, -gs, -ys, -ds, -ws, -ls, -rs, -ks, -en\}$ 
10. for all  $V \in L_{UV}$  do
11.    $N \leftarrow freqCount(V)$ 
12.   if  $N \geq \theta$  then //  $\theta$  is a threshold value
13.      $L_{RV} \leftarrow L_{RV} \cup V$ 
14.   end if
15. end for
16.  $R \leftarrow L_{RV}$ 
17. for all  $V_i \in L_{RV}$  do // identifying morphological variants
18.   for all  $V_j \in L_{UV}$  do
19.     if  $V_i \in subString(V_j)$  then
20.        $R \leftarrow R \cup V_j$ 
21.     end if
22.   end for
23. end for
24. Return  $R$ 

```

Table 1.5 A partial list of feasible biological relations and their morphological variants

Biological relations	Morphological variants
associate	associate with, associated with, associated to
increase	increased, increases, increased in, increased after, increased by, increased over
induce	induced, induced by, induces, induced in, induced with
show	showed, shown, shown on, show for, shows
reduce	reduced, reduces, reduced by, reduced in
decrease	decreased in, decreased as, decreased with, decreased across
regulate	regulated by, regulates
affect	affected, affects, affected in, affected by, affecting
express	expressed in, expressing, express as, expresses, expressed from
attenuate	attenuated, attenuated by, attenuates, attenuated in
generate	generated by, generated from
enhance	enhanced in, enhanced by
activate	activates, activated
inhibit	inhibits, inhibited, inhibited with, inhibition, inhibited by
modulate	modulates, modulated, modulated in, modulated by
stimulate	stimulates, stimulated, stimulated with, stimulated by

1.4 PERFORMANCE EVALUATION

The performance of the system is analyzed by taking into account the performance of the biological relation extraction process, which aims to identify relevant verbs signifying biological entity interactions from MEDLINE abstracts. We have already explained the extraction process in the previous sections. We now present detailed discussion about how we evaluate the correctness of the extracted biological relations through analyzing the original sentences in which these relational verbs occur. In order to evaluate the correctness of the extraction process, we have randomly selected 10 different feasible biological relations and 100 GENIA abstracts for manual verification. The entity markers were removed from the GENIA abstracts before applying our relation mining algorithm.

A biological relation is said to be *correctly identified* if its occurrence within a sentence along with its left and right entities is grammatically correct and the system has been able to locate it in the right context. To judge the performance of the system, it is not enough to judge the extracted relations only, but it is also required to analyze all the correct relations that were missed by the system. The system is evaluated for its *precision*, *recall* and *F-score* values by considering 10 relations – *activate*, *associate*, *express*, *increase*, *induce*, *inhibit*, *modulate*, *reduce*, *regulate* and *stimulate*. For evaluation of the system, an evaluation software was written in *Java*, which exhaustively checks the corpus for possible occurrences of the required relation. For each relation to be judged, the evaluation software takes the root relation as input and performs partial string matching to extract all possible occurrences of the relation. This ensures that various nuances of English language grammar can also be taken care of. For example, if the root relation used in any query is “*activate*”, all sentences containing *activates*, *inactivate*, *activated by*, *activated in*, etc. are extracted. Each sentence containing an instance of the pattern is presented to the human evaluator after its appropriate tagging through ABNER. The sentence without ABNER tags is also presented to the evaluator. This makes it easier for the evaluator to judge the grammatical correctness of the relation in association to the concepts or entities around it. Each occurrence of the relation is judged for correctness by the evaluator, and the correct instances are marked. The marked instances are stored by the evaluation software and later used for computing the precision (π), recall (ρ) and *F-score* (F_1) values by using equations 1.1, 1.2 and 1.3, respectively.

The precision value of the system reflects its capability to identify a relational verb along with the correct pair of concepts/entities within which it is occurring. Recall value reflects the capability of the system to locate all instances of a relation within the corpus. Table 1.6 summarizes the performance measure values of our relation mining system in the form of a misclassification matrix for information components centered around 10 different biological relations. On 100 randomly selected documents from GENIA corpus, the average precision, recall, and *F-score* values are 92.71%, 73.07%, and 81.73% respectively.

$$precision(\pi) = \frac{TP}{TP + FP} \quad (1.1)$$

$$recall(\rho) = \frac{TP}{TP + FN} \quad (1.2)$$

$$F\text{-score}(F_1) = 2 \times \frac{\pi \times \rho}{\pi + \rho} \quad (1.3)$$

Table 1.6 Evaluation results of the biological relation extraction system

Biomedical relation	No. of times IC is identified by the system	No. of times IC is correctly identified by the system	No. of times IC occurs correctly in the text corpus	π (%)	ρ (%)	F_1 (%)
Activate	36	35	49	97.22	71.43	82.35
Associate	19	18	22	94.74	81.82	87.80
Express	26	24	35	92.31	68.57	78.69
Increase	19	17	26	89.47	65.38	75.56
Induce	71	67	91	94.37	73.63	82.72
Inhibit	36	34	48	94.44	70.83	80.95
Modulate	6	5	6	83.33	83.33	83.33
Reduce	22	21	30	95.45	70.00	80.77
Regulate	31	28	37	90.32	75.68	82.35
Stimulate	22	21	30	95.45	70.00	80.77
Average				92.71	73.07	81.73

As is observed, the precision of the system is quite high. This indicates that most of the extracted instances are correctly identified. However, the recall value of the system is somewhat low. This indicates that several relevant elements are not extracted from the text. The reason for low recall values is identified as follows. We observed that most miss occur when the parser assigns an incorrect syntactic class to a relational verb. For example, in the following sentence, the relational verb *activates* and other related constituents could not be identified by the system because *activates* is marked as noun by the parser. Similarly, other misses occur when an information components spans over multiple sentences using anaphora.

“Increased [Ca²⁺]_i activates Ca²⁺/calmodulin-dependent kinases including the multifunctional Ca²⁺/calmodulin-dependent protein kinase II (CaM-K II), as well as calcineurin, a type 2B protein phosphatase [MEDLINE ID: 95173590]”

1.5 UNIQUENESS OF THE PROPOSED BIOLOGICAL RELATION MINING SYSTEM

The primary focus of the proposed biological relation mining system is to locate complex information components embedded within non-annotated biomedical texts, where an information component comprises biological concepts and relations. Though a number of systems have attempted to do the same task, there are certain unique aspects to the proposed approach, which we highlight in this section. The proposed text-mining based approach unifies natural language processing and pattern mining techniques to identify all feasible biological relations within a corpus. Unlike most of the related work [20, 21, 22, 26], that have described methods for mining a fixed set of biological relations occurring with a set of predefined tags, the proposed system identifies all verbs in a document, and then identifies the feasible biological relational verbs using contextual analysis. While mining biological relations the associated prepositions are also considered which very often changes the nature of the verb. For example, the relation *activates in* denotes a significant class of biological reactions. Thus, we also consider the biological relations, which are combinations of root verbs, morphological variants, and prepositions that follow these. Typical examples

of biological relations identified in this category include *activated in*, *binds to*, *stimulated with*, etc. Besides mining relational verbs and associated entities, the novelty of the system lies in extracting validatory entities whose presence or absence validates a particular biological interaction. The system also extracts the adverbs associated with relational verbs, which plays a very important role especially to identify the negation in sentences that are very crucial while answering biomedical queries. Unlike the related work [5], which have described method for mining biological relations from tagged GENIA corpus, the proposed system has been designed to work with a collection of untagged biomedical literature.

1.6 CONCLUSION AND FUTURE WORK

In this chapter, we have presented how text mining can be extended to extract generic biological relations from text corpus. The system uses linguistic and semantic analysis of text to identify NP and VP phrases and their semantic relations to represent texts using conceptual graphs, which are then analyzed to identify relation instances and map them into information components. The information components are centered on domain entities and their relationships, which are extracted using natural language processing techniques and co-occurrence-based analysis. The proposed system employs text mining principles along with NLP techniques to extract information about the likelihood of various entity-relation occurrences within text documents. Though the system design is fairly generic, the design of the entire system has been validated with experiments conducted over PubMed abstracts. Performance evaluation result shows that the precision of the relation extraction process is high. Reliability of the process is established through the fact that all manually identified relational verbs are extracted correctly. The recall value however may be improved with more rigorous analysis of the phrase structure tree generated by the parser. Extracted feasible biological relations along with information components can be used for knowledge visualization and efficient information extraction from text documents to answer biomedical queries posted at different levels of specificity.

One of the interesting applications of the conceptual graphs, generated as an intermediate representation of the texts, is to identify biological relations associations at generic concept-levels, rather than at entity-level by using the *GP-Close* algorithm proposed in [4] for mining frequent generalized association patterns. For this, we may utilize the concept hierarchies defined in existing biological ontologies (e.g., GENIA ontology [17]) to map extracted biological entities from texts over them and then to characterize biological relations at concept-levels. Presently, we are enhancing our system to incorporate *GP-Close* algorithm to mine frequent generalized association for the identified generic biological relations. This could be very helpful to enhance existing biological ontologies using generic relations mined from biological text documents.

REFERENCES

1. A.-H. TAN. Text Mining: The State of the Art and the Challenges. In *Proceedings of the Pacific Asia Conference Knowledge Discovery and Data Mining (PAKDD '99) Workshop Knowledge Discovery from Advanced Databases*, pages 65–70, 1999.

2. J. DORRE, P. GERSTL AND R. SEIFFERT. Text Mining: Finding Nuggets in Mountains of Textual Data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 398–401, 1999.
3. N. GUARINO, C. MASOLO AND G. VETERE. Ontoseek: Content-Based Access to the Web. *IEEE Intelligent Systems*, **14**[3]: 70–80, 1999.
4. T. JIANG, A.-H. TAN AND K. WANG. Mining Generalized Associations of Semantic Relations from Textual Web Content. *IEEE Transactions on Knowledge and Data Engineering*, **19**[2], 2007.
5. M. ABULAIISH AND L. DEY. Biological relation extraction and query answering from medline abstracts using ontology-based text mining. *Data and Knowledge Engineering*, **61**[2]:228–262, 2007.
6. S. ALBERT, S. GAUDAN, H. KNIGGE, A. RAETSCH, A. DELGADO, B. HUHSE, H. KIRSCH, M. ALBERS, D. R. SCHUHMAN, AND M. KOEGL. Computer-assisted generation of a protein-interaction database for nuclear receptors. *Molecular Endocrinology*, **17**[8]:1555–1567, 2003.
7. J. ALLEN. *Natural Language Understanding*. Pearson Education (Singapore) Pvt. Ltd., Indian branch, 2nd edition, 2004.
8. M. BERARDI, D. MALERBA, R. PIREDDA, M. ATTIMONELLI, G. SCIOSCIA, AND P. LEO. *Biomedical Literature Mining for Biological Databases Annotation*, chapter 16, pages 320–343. I-Tech, Vienna, Austria, 2008.
9. A. BERNSTEIN, E. KAUFMANN, A. GOHRING, AND C. KIEFER. Querying ontologies: A controlled english interface for end-users. In *Proceedings of the International Semantic Web Conference*, pages 112–126, 2005.
10. M. CIARAMITA, A. GANGEMI, E. RATSCH, J. SARIC, AND I. ROJAS. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 659–664, 2005.
11. 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. *Bioinformatics*, **17**[1]:S74–S82, 2001.
12. K. FUNDEL, R. KUFFNER, AND R. ZIMMER. Relex - relation extraction using dependency parse trees. *Bioinformatics*, **23**:365–371, 2007.
13. R. GAIZAUSKAS, G. DEMETRIOU, P. J. ARTYMIUK, AND P. WILLETT. Protein structures and information extraction from biological texts: the pasta system. *Bioinformatics*, **19**[1]:135–143, 2003.
14. D. GAVRILIS, E. DERMATAS, AND G. KOKKINAKIS. Automatic extraction of information from molecular biology scientific abstracts. In *Proceedings of the International Workshop on Speech and Computer (SPECOM'03)*, 2003.
15. L. HIRSCHMAN, A. YEH, A. MORGAN, AND M. COLOSIMO. Linking biological literature, information, and knowledge. *The EDGE - MITRE's Advanced Technology Newsletter*, **9**[1]:8–9, 2005.
16. T. K. JENSSEN, A. LAEGREID, J. KOMOROWSKI, AND E. HOVIG. A literature network of human genes for high-throughput analysis of gene expression. *Nature Geneics*, **28**:21–28, 2001.
17. J. D. KIM, T. OHTA, Y. TETEISI, AND J. TSUJII. GENIA ontology. Technical Report TR-NLP-UT-2006-2, Tsujii Laboratory, University of Tokyo, 2006.
18. M. MIWA, R. SAETRE, Y. MIYAO, AND J. TSUJII. Protein-protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, **78**[12]:39–46, 2009.

19. S. MUKHERJEA AND S. SAHAY. Discovering biomedical relations utilising the world-wide-web. In *Proceedings of the 11th Pacific Symposium on Biocomputing, Hawaii*, pages 164–75, 2006.
20. T. ONO, H. HISHIGAKI, A. TANIGAMI, AND T. TAKAGI. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, **17**[2]:155–161, 2001.
21. F. RINALDI, G. SCHEIDER, C. ANDRONIS, A. PERSIDIS, AND O. KONSTANI. Mining relations in the GENIA corpus. In *Proceedings of the 2nd European Workshop on Data Mining and Text Mining for Bioinformatics, Pisa, Italy*, pages 61–68, 2004.
22. T. SEKIMIZU, H. S. PARK, AND J. TSUJII. Identifying the interaction between genes and genes products based on frequently seen verbs in medline abstract. *Genome Inform*, **9**:62–71, 1998.
23. B. SETTLES. ABNER: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, **21**[14]:3191–3192, 2005.
24. P. SRINIVASAN. Text mining: Generating hypotheses from medline. *Journal of the American Society for Information Science*, **55**[4]:396–413, 2004.
25. B. J. STAPLEY AND G. BENOIT. Bibliometrics: Information retrieval and visualization from co-occurrence of gene names in medline abstracts. In *Proceedings of the 5th Pacific Symposium on Biocomputing, Hawaii*, pages 529–540, 2000.
26. J. THOMAS, D. MILWARD, C. OUZOUNIS, S. PULMAN, AND M. CARROLL. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the 5th Pacific Symposium on Biocomputing, Hawaii*, pages 538–549, 2000.
27. Y. TSURUOKA, Y. TATEISHI, J. D. KIM, T. OHTA, J. MCNAUGHT, S. ANANIADOU, AND J. TSUJII. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics – 10th Panhellenic Conference on Informatics*, pages 382–392, 2005.
28. T. WATTARUJEEKRIT, P. K. SHAH, AND N. COLLIER. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, **5**:155–174, 2004.
29. J. D. WREN AND H. R. GARNER. Shared relationship analysis: Ranking set cohesion and commonalities within a literature-derived relationship network. *Bioinformatics*, **20**[2]:191–198, 2004.
30. Y. XU, Z. CHANG, W. HU, L. YU, H. DUANMU, AND X. LI. Mining the relationship between gene and disease from literature. In *Proceedings of the 6th International Conference on Fuzzy System and Knowledge Discovery (FSKD'09), Tianjin*, pages 482–486, 2009.
31. PubMed home page, <http://www.ncbi.nlm.nih.gov/pubmed>
32. U.S. National Library of Medicine (NLM), <http://www.nlm.nih.gov/>
33. Medline home page, <http://www.nlm.nih.gov/pubs/factsheets/medline.html>
34. MeSH home page, <http://www.nlm.nih.gov/mesh/>
35. World Wide Web Consortium, <http://www.w3.org/>
36. Stanford's Parser, <http://nlp.stanford.edu/downloads/lex-parser.shtml>