A protocol for constructing a domain-specific ontology for use in biomedical information extraction using lexical-chaining analysis

by

Xiaofen He

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2006

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In order to do more semantics-based information extraction, we require specialized domain models. We develop a hybrid approach for constructing such a domain-specific ontology, which integrates key concepts from the protein-protein–interaction domain with the Gene Ontology. In addition, we present a method for using the domain-specific ontology in a discourse-based analysis module for analyzing full-text articles on protein interactions. The analysis module uses a *lexical chaining* technique to extract strings of semantically related words that represent the topic structure of the text. We show that the domain-specific ontology improved the performance of the lexical-chaining module. As well the topic structure as represented by the lexical chains contains important information on protein-protein interactions appearing in the same textual context.

Acknowledgments

The completion of this thesis was possible only with the assistance of several very supportive and kind individuals. First I must thank my supervisor, Dr. Chrysanne DiMarco. Dr. DiMarco's help and friendship throughout my time at the University of Waterloo has been tremendously valuable and is greatly appreciated. Shady, Matttew, Aaron, Gabe and Zhou, they have given great help to my studies. My husband, Xiaoyang, and my daughter, Michelle, have been my principal source of inspiration and support. I could not have completed my work without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Each living cell is rich in proteins that continuously interact with each other. Knowledge about the identities and functions of interacting proteins contributes significantly to the understanding of biological processes by providing insight into the roles of important genes, elucidating relevant pathways, and facilitating the identification of potential drug targets for use in developing novel therapies.

A large volume of protein-protein interactions has been identified, and information about such interactions is now readily available in online databases such as BIND [6]. However, the information stored in current databases does not allow us to rank the biological validity of the interactions—it may be the case that interactions occurring under laboratory conditions do not actually occur in the living cell. A researcher trying to establish the quality of the interactions identified in a database could read the details of the experiments in each related scientific article, but this is labourious and time-consuming. If the number of relevant papers is high, it will be difficult or even impossible for a researcher to manually process all the articles to assess the value of the interactions. For example, a text query in BIND for interactions of the single protein *Cdc42* will retrieve 512 records, far too many to be easily read and analyzed by manual methods—there is a clear need for an automated information extraction system to assist researchers in analyzing the online literature to better judge the quality of protein-protein interactions.

We set out to develop such an automated information extraction system that uses a Natural Language Processing (NLP) discourse analysis technique: *lexical chaining*. The notion of lexical chaining derives from the concept of textual cohesion. A *lexical chain* is a sequence of semantically related words in the text, spanning a topical unit of the text, i.e., a set of adjacent words or sentences, or the entire text. Lexical chaining is the process of extracting and connecting semanti-

cally related words from a text, then creating a set of word chains that represent the various topics throughout the text. A core part of the lexical-chaining technique is the lexical knowledge resource used for determining the 'semantic relatedness' of words.

As the molecular biology literature provides detailed descriptions of protein interaction experiments specifying the individual interaction partners, as well as the corresponding interaction types, it has been exploited as a resource to derive protein interactions for interaction databases. We hoped to extract similar information to confirm or judge the biological quality of the protein interactions. However, we found that a major barrier to our goal is the lack of a readily available lexical knowledge resource that covers vocabulary in protein-protein–interaction domains. The most widely used lexical knowledge resource for current lexical-chaining algorithms is WordNet [3], but WordNet represents only the general English lexicon. We then explored the use of the most well-known biological ontology, the Gene Ontology [18]. The Gene Ontology has a hierarchical structure similar to WordNet, which makes it easy to convert into a WordNet-like lexical knowledge database. However, the Gene Ontology lacks protein-protein–interaction domain-specific information, even though it has a broad vocabulary for general biological domains.

In order to extract valuable protein interaction information that can be used to judge the quality of protein-protein interactions, it is necessary for us to build a protein-protein–interaction domain-specific ontology. Two types of approaches have been used for building ontologies: manual methods and automated methods. An example of a manual method is the one proposed by Uschold and King [49]. Their method involved three stages: identifying the purpose of the ontology (i.e., why to build it, how it would be used, range of the users), building the ontology, evaluation and documentation. Automated methods often rely on corpus-based statistical approaches that automatically build domain-specific concept structures. Typically, an automatic approach first selects suitable text corpora to represent the domains of interest, then finds statistical evidence about terms in the text corpora, then finally determines relationships between concepts by considering statistical evidence in text corpora.

This study explores a hybrid approach that combines the corpus-based statistical method that extracts domain-specific concepts from a protein-protein–interaction (PPI) text corpus semi-automatically, and a manual method that builds a domain-specific ontology using the concepts extracted. The PPI domain-specific ontology was then integrated into the Gene Ontology. We ran experiments using lexical-chaining analysis to extract information from the protein-protein–interaction literature using the original Gene Ontology and our expanded PPI-specific Gene On-

tology. The experiment results clearly indicate that the additional PPI ontology has a very positive impact on the quality and quantity of information extracted.

This thesis makes several significant contributions to the study of biomedical information extraction. We develop an innovative approach to biomedical information extraction that uses NLP discourse analysis based on a lexical-chaining technique. We present a hybrid methodology for constructing a domain-specific ontology for use in lexical-chaining analysis, and use our methodology to develop a protein-protein–interaction ontology extension to the Gene Ontology. We provide experimental results that indicate the use of a domain-specific ontology improves the performance of information extraction based on lexical-chaining analysis. We also investigate several metrics for lexical chains and the possibility of using these metrics to judge the biological validity of protein-protein interactions.

This thesis consists of six chapters. Chapter 1 introduces the motivation and methodology for our study. Chapter 2 reviews relevant research in biomedical information extraction, including the state-of-the-art of systems using NLP techniques. In Chapter 3, we introduce the lexical-chaining technique and present the details of the lexical-chaining algorithm we adopted in our study. In Chapter 4, we describe our methodology for constructing a domain-specific ontology for use in the lexical-chaining analysis module. In Chapter 5 we conduct an experimental evaluation of the constructed domain-specific ontology, and compare its performance in analyzing protein-protein–interaction texts to an existing biological ontology, the Gene Ontology. The Chapter 6 concludes the thesis with a summary and suggestions for future work.

Chapter 2

Survey of Biomedical Information Extraction

2.1 Information Extraction

Information Extraction (IE) is the process of extracting information from natural-language text, one of the most prominent techniques currently used in *Text Mining*. Text mining, as Hearst [27] put it, is "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources". Text mining is different from traditional *Information Retrieval (IR)*. Information Retrieval focuses more on the larger units of text such as documents, and usually the information retrieved is delivered in the form of complete documents. For example, an information retrieval task could be helping users find documents that satisfy their information needs. On the other hand, text mining is also different from pure *Natural Language Processing (NLP)*, even though some natural language processing techniques are widely used in text mining. Natural language processing is a general description for all attempts to use computers to process the languages naturally used by humans. It aims to understand the meaning of the whole text, while text mining focuses more on solving a specific problem at a time, i.e., identifying needed information, detecting certain relationships of interest, and so on [17]. Information extraction has been particularly useful in text mining tasks. Rather than mining 'a nugget of gold' from a sea of irrelevant information, Information Extraction aims to assemble/discover new knowledge from vast amount of texts when none of these is particularly valuable alone.

In order to evaluate system performance, developers of information extraction applications have adopted several standard evaluation metrics from information retrieval including *precision*, *recall*, and a combined metric, *F-measure* [30]. Recall measures how much relevant information the system has extracted from the text; precision measures how much of the information that the system returned is actually correct. Often these two measures are antagonistic to one another, and the decision as to which measure is more important may be dependent on the application. F-measure balances recall and precision by using a weight parameter β. The formulas to calculate the three measures are listed below:

$$R = recall = \frac{\#\ of\ correct\ answers\ given\ by\ system}{total\ \#\ of\ possible\ correct\ answers\ by\ system}$$

$$P = precision = \frac{\#\ of\ correct\ answers\ given\ by\ system}{\#\ of\ answers\ by\ system}$$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

Biomedical information extraction has attracted increasing attention in recent years. The volume and complexity of published biomedical research is expanding at an impressive and even intimidating rate. As of 2006, MEDLINE[1] contains over 15 million bibliographic citations from more than 5,000 biomedical journals worldwide; over 623,000 total references were added in 2006 alone.

It is difficult and often impossible for researchers to find what they are looking for within this huge sea of data. For example, researchers trying to assess the biological validity of interactions of a protein could read the details of the experiments in each related scientific article, but a text query in a protein-protein interaction database such as BIND [6] for interactions of the single protein Cdc42 will retrieve 512 records, far too many to be easily read and analyzed by manual methods. There is a clear need for automated information extraction solutions to assist researchers in processing and analyzing biomedical literature. Biomedical information extraction has focused on the following three tasks:

Named Entity Recognition:

Identifying gene and protein names in biomedical text.

[1]MEDLINE is the National Library of Medicine's premier bibliographic database covering the fields of medicine, nursing, dentistry, veterinary medicine, the health care system, and the preclinical sciences.

Named Entity Normalization:
Mapping genes to their unique identifiers in organism databases.

Functional Annotation:
Associating proteins/genes that interact with one other.

Among many information extraction techniques, NLP techniques are now widely used in biomedical information extraction. A typical NLP approach usually involves a simple recognition of basic grammatical features (e.g., each word's part-of-speech), and then a 'shallow' syntactic analysis using targeted grammatical rules to identify elemental units (e.g., noun phrases, verb phrases) within the sentence.

A detailed survey of representative work on these specific tasks will be given later in this chapter. First we will describe two systems, GENIES [22] and PASTA [24], to demonstrate the utility of NLP techniques in biomedical information extraction.

2.1.1 GENIES

GENIES (Genomics Information Extraction System) [22] is a component of a comprehensive information extraction system called *GeneWays*. The creators of GeneWays had ambitious goals to perform massive automated extraction, analysis, visualization, and integration of molecular pathway data. As the core component of GeneWays's NLP module, GENIES' responsibility is to extract and structure information related to molecular pathways by parsing full-text articles collected from the websites of scientific journals.

The architecture of GENIES is shown in Fig 2.1. GENIES consists of six components: two internal knowledge sources (Lexicon and Grammar) and four processing components (Term Tagger, Preprocessor, Parser, and Error Recovery).

The manual construction of the knowledge sources takes significant effort. The creators of GENIES manually developed an ontology for the signal transduction domain and built a semantic grammar along with manually derived syntactic and semantic constraints [42].

The first step in text processing is the part-of-speech tagging of the input text. The Term Tagger [32] used BLAST techniques, specialized rules, and external knowledge sources, such as GeneBank [10] and Swiss-Prot [7], to identify and tag genes and proteins. BLAST (Basic Local Alignment Search Tool) [5] is a heuristic algorithm that attempts to optimize the process of DNA and protein sequence comparison. The Term Tagger made use of BLAST by mapping sequences of text characters into sequences of nucleotides, which can be processed by BLAST.

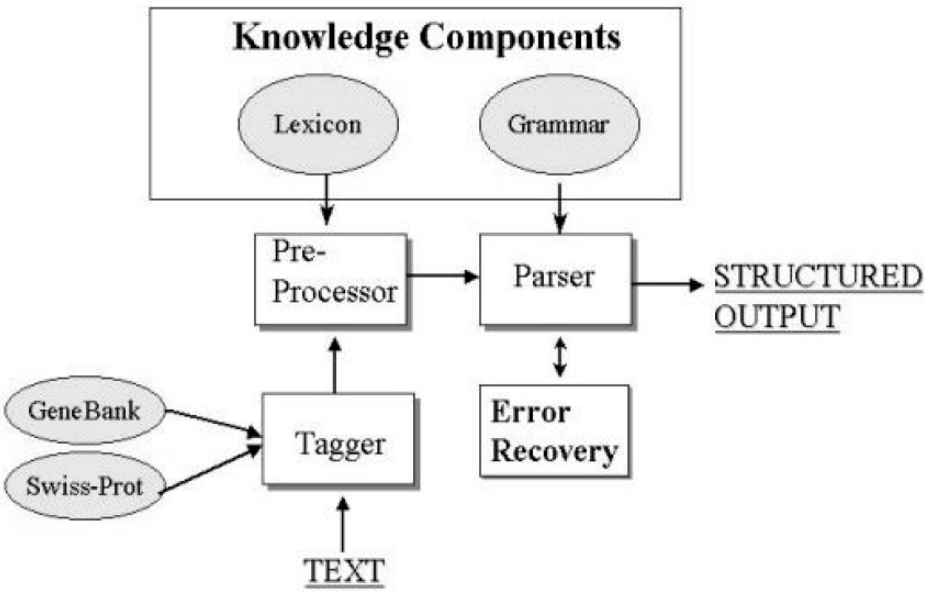


Figure 2.1: The architecture of GENIES [22]

Secondly, the Preprocessor segments articles into sentences, words, or atomic multi-word phrases. Words and phrases that are not tagged in the previous step can be further identified through lexicon lookup.

Thirdly, the Parser uses grammatical rules to recognize, fill semantic patterns, and generate target output. GENIES' target output can be viewed as a list of elements and relations that link the elements. Elements are tagged with type and value. Relations usually are more complex and in some cases nested. As an illustration, the sentence *"phosphorylated Cbl coprecipitated with Crkl, which was constitutively associated with the C3G"* will generate the following target output:

[action, attach,
[protein, Cbl, [state, phosphorylated]],
[protein, Crkl, [action, *attach*, [protein, *Crkl*], [protein, *C3G*]]]]

In the output, there is one nested relation (*attach* in line 1) and one primary relation (*attach* in line 3), corresponding to *coprecipitated with* and *associated with* respectively; the primary *attach* relation links protein element *Crkl* with *C3G*, while

the nested relation links *Cbl* with *Crkl*.

Lastly, the Error Recovery module uses various strategies to parse sentence components to improve precision.

GENIES' performance was measured against human experts. After processing a 8000-word article selected from Cell[2], GENIES obtained 96% precision and 53% recall. With this satisfyingly high precision, GENIES shows promising capabilities to extract valuable and complex information from biomedical text. However, the evaluation was only performed on a single article, so that more thorough evaluation needs to be done. GENIES also only extracts information on a single-sentence basis. In the next section, we will describe another system, PASTA [24], which deals with multiple sentences.

2.1.2 PASTA

Gaizauskas et al. [24] described the *PASTA (Protein Active Site Template Acquisition)* project, whose goal is to extract detailed information of the roles of amino acids in protein molecules, to place the information into structured representations, and to generate a database of protein-active sites from both scientific journal abstracts and full articles. PASTA uses a pipeline architecture consisting of four principal stages:

Text preprocessing.

This stage consists of three activities: section analysis, tokenization, and sentence splitting.

1. Section analysis: Use a set of regular expressions to identify those sections in a text that are considered relevant for information extraction.
2. Tokenization: Segment the relevant text sections into the smallest processing atoms. For example, word *Cys128* will have two tokens: *Cys* and *128*; compound protein name *casein kinase* will also have two tokens: *casein* and *kinase*.
3. Sentence splitting: Segment the text into sentences.

Terminological Processing.

This stage contains three modules: morphological processing, lexical lookup, and terminology parsing. A protein name *casein kinase* is used as an example to illustrate each module's functionality.

[2]http://www.cell.com/

1. Morphological processing: Identify tokens that contain interesting biochemical affixes such as *-ase* or *-in*. Token *kinase* thus may be identified as a 'protein_head' term based on the morphological affix *-ase*, while *casein* may be identified as 'protein_modifier' term based on the affix *-in*.
2. Lexical lookup: Use a series of finite state recognizers to identify and if possible classify the token-sequences. Token *casein* and *kinase* may be identified as a compound term based on their position in the text.
3. Terminology parsing: Use a rule-based terminology parse to analyze, assemble, and classify the identified token/token sequences to single multi-token unit. *casein kinase* will be recognized as a protein name by a grammatic rule such as:

   protein –>protein_modifier, protein_head, numeral.

Syntactic and Semantic Processing.

At this stage, text is processed on a sentence-by-sentence basis. Each sentence is transformed into a semantic representation by applying the NLP syntactic analysis (part-of-speech tagging and phrasal parsing) followed by the transduction of grammatical form into a predicate-argument semantic representation. As an illustration, the predicate-argument semantic representation derived from the sentence *Ser154, Tyr167 and Lys171 are found at the active site* is shown in Table 2.1.

Discourse Processing and Template Extraction.

At this stage, the semantic representations from multiple sentences are linked by making inferences using a predefined domain model, which is made up of a concept hierarchy, inheritable properties of concepts, and inference rules associated with concepts. Then the linked semantic representations are further merged/added into the domain model. Following the discourse process, a template-writing module scans the final domain model for information relevant to the templates and eventually generates filled templates.

The preliminary experiments for the PASTA system achieved an average of 94% precision and 88% recall for terminology recognition and classification on a test set of 52 abstracts.

Although there was a lack of thorough evaluation, PASTA and GENIES were among the first systems that demonstrated the feasibility of automatically building a structured knowledge base directly from the literature using NLP techniques with

Predicate	Arguments
residue(e1)	name(e1,'Ser154')
residue(e2)	name(e2,'Tyr167')
residue(e3)	name(e3,'Lys171')
set(e4)	member(e4,e1), member(e4,e2), member(e4,e3)
find(e5)	lobj(e5,e4)
active_site(e6)	at(e5, e6)

Table 2.1: An example of the predicate-argument representation [24]

promising test results. In the following sections, we will survey the achievements and efforts for the major Information Extraction tasks.

2.2 Named Entity Recognition and Normalization

One of the initial challenges of Information Extraction systems is to recognize the entity names in the text. This task includes two steps: *Named Entity Recognition (NER)* and *Named Entity Normalization (NEN)*. Named Entity Recognition is the identification of text terms that refer to concepts of interest in specific domains, whereas Named Entity Normalization is the mapping of these terms to the unique concept to which they refer. The targeted entities in the biomedical domains include genes, proteins, chemicals, cells, and organisms, etc.

2.2.1 Named Entity Recognition

The Named Entity Recognition problem has attracted extensive attention and many techniques have been well-developed, but it still remains a challenging task in biomedical domains. This is largely because the entity names in biomedical domains are much more complex than in other domains. The naming conventions in biomedical domains have the following characteristics ([52], [35]):

Descriptive naming convention.
Without the standardized gene-naming rules in biology, biomedical entity names are often derived from descriptive terms and vary considerably in style

from organism to organism. Moreover, the conventions for new gene/protein names often depends partially on the author's style. The descriptive style of naming makes it difficult to identify the left boundaries of such names.

Conjunction and disjunction.
Two or more biomedical entity names may share one head noun due to conjunction or disjunction, e.g., *'91 and 84 kDa proteins'* consists of two entity names: *'91 kDa proteins'* and *'84 kDa proteins'*.

Unknown words.
With the explosive growth in the volume of biomedical literature, new entity names are being created constantly, and only later will be recognized by domain experts through repetition of use. This consequently results in the low coverage of existing biomedical dictionaries.

Acronyms and Abbreviation.
Acronyms and abbreviations are frequently used in biomedical domains. Because of the ambiguity of acronyms and abbreviations that refer to multiple terms, sometimes across multiple domains, the classes of acronyms or abbreviations cannot be resolved by use of dictionaries alone, and are very much dependent on the context.

Cascaded construction.
A biomedical entity name may be embedded in another biomedical entity name. Consider the named entity *kappa 3 binding factor*. Its annotation *<PROTEIN><DNA>kappa 3 </DNA>binding factor </PROTEIN>* has two right boundaries at *3* and *factor*, which correspond to the embedded named entity in the DNA category and the nested named entity of the Protein category, respectively.

There are three basic approaches to Named Entity Recognition: rule-based, dictionary-based, and context-based. In a rule-based approach, entity names are extracted by applying a set of manually developed rules that exploit surface cues and use simple linguistic and domain knowledge ([23], [37]). In a dictionary-based approach, a long list of patterns that cover terms from the dictionary and their variations is first constructed. The text is tokenized and each textual n-gram segment in the text is scanned for matches to the patterns in the list [31]. The context-based approach is the most popular one. With this approach, the recognizer is trained on an annotated corpus by using statistical machine-learning techniques, such as Hidden Markov Models (HMM) [41], Support Vector Machines (SVM) [19], etc.

The machine-learning classifier is used to determine the text regions corresponding to entity names.

Each of these approaches has its advantages and disadvantages, but they need not be used in isolation. Mika and Rost [35] constructed a system that combined dictionary-based and rule-based filtering with several SVMs to identify protein names in MEDLINE abstracts. We will describe an example of each approach below.

2.2.1.1 Fukuda et al. (1998)

Fukuda et al. [23] pioneered the automated identification of protein names. Before Fukuda, the best-known strategy was to prepare proper-noun dictionaries and a syntactic pattern dictionary. However, the performance of this strategy depended heavily on the quality of the proper-noun dictionaries used so that preprocessing was necessary in which patterns were used to extract compound words as a word. Fukuda et al. proposed a method, *PROPER (PROtein Proper-noun phrase Extracting Rules)*, which extracts entity names using surface cues on character strings in biomedical documents. This method uses the characteristics of proper-noun descriptions in these research fields, and does not require pre-existing dictionaries of proper nouns. For example, protein names are classified into three categories according to their properties such as the occurrences of uppercase letters, numerals and special endings, etc. As a result, PROPER can extract names with high accuracy, regardless of whether the name is a known/unknown word or a single/compound word.

In this study, the targeted entity names included protein names, protein domain names or motifs, sites, fragments, and elements, etc. Fukuda et al. further defined two types of terms, *core-terms* and feature-terms, to categorize individual words. Core-terms are words which provide core information such as protein names and have recognizable surface features, such as capital letters, numerals, special symbols, so that they can be clearly distinguished from general words. Feature-terms describe the domain-specific functions and characters of compound words, and can be used for classification of the compound words. In the following phrase, *EGF receptor*, *EGF* is the core-term, and *receptor* is the feature-term. The process flow of PROPER is summarized as follows:

1. The text is split into sentences, tokenized, and then tagged using a part-of-speech tagger.

2. Next, all words which are syntactically predicted to be a core-term are extracted as *candidate words*, e.g., words with uppercase, numerical figures, etc. Words which are semantically unacceptable as core-terms are then removed from the candidate word list.

3. Adjacent core-terms and feature-terms are concatenated by matching against surface rules and/or part-of-speech rules. In this step, noun phrases without conjunctions and prepositions are reconstructed.

4. Noun phrases generated in the last step are further combined at the sentence basis by applying several simple patterns.

5. Improper part-of-speech annotations are removed to achieve high recall.

The system was evaluated on 30 MEDLINE abstracts on *SH3* domain and 50 MEDLINE abstracts for *signal transduction*. The results showed precisions on various levels ranging from 90% to 96% with recall roughly in the same range as well.

2.2.1.2 Narayanaswamy et al. (2003)

Narayanaswamy et al. [37] described a name entity extraction system that was inspired by Fukuda's method. They improved upon Fukuda's method in two ways: first, they improved the precision and recall significantly; secondly, they recognized not only protein and gene names, but also other types of names. Their target entity names included protein/gene names, protein/gene parts, chemical names, chemical parts, source terms (e.g., cells, cell parts, organisms, etc.), and general biological terms that could not be classified into the above classes.

The approach taken was symbolic, based on a set of manually developed rules. These rules exploited surface cues and simple linguistic and domain knowledge to identify the relevant terms in the biomedical literature.

Like Fukuda's method, Narayanaswamy et al. categorized two types of terms, *core-terms* and *functional-terms*, and proceeded to identify these terms using similar procedures. However they differed from Fukuda et al. in the following aspects:

Classification.
While Fukuda et al. did not deal with classification, Narayanaswamy et al. associated each extracted term with its classification (protein/gene, protein/gene parts, chemical, chemical parts, and source).

Abbreviations.

A simple algorithm was used to identify abbreviations and associate each identified abbreviation with its classification. Their algorithm was based on the observation that the first occurrence of an abbreviation typically occurs within the parenthesis following the original term.

Core-terms.

Two types of core-terms were further defined: protein c-terms and chemical c-terms. General core-terms are extracted by applying surface rules similar to Fukuda et al. Among the general core-terms, chemical c-terms were identified through the recognition of chemical root forms based on the *International Union for Pure and Applied Chemistry (IUPAC)* chemical naming conventions and other morphological features such as suffixes. Protein c-terms are identified solely on the basis of suffixes such as *-ase*.

They evaluated the system on 55 MEDLINE abstracts collected by searching for *acetylates*, *acetylated*, and *acetylation*. The test set had a good proportion of protein, protein part, and chemical names. The evaluation showed precision, recall, and F-measure values of 90.39%, 95.64%, and 92.94%, respectively. The authors also ran Fukuda's system on their test set. They claimed that their system substantially outperformed Fukuda's, and that the difference in precision could be largely attributed to the presence of chemical and source names.

2.2.1.3 Kou et al. (2003)

Kou et al. [31] proposed a novel dictionary-based Named Entity Recognition method that was part of a larger image and text extraction system, SLIF. Their systems's characteristics required focusing on identification of entities from a fixed list that could change over time and considering recall to be more important than precision.

Their method was a novel approach that combined a dictionary-based method with a statistical machine-learning method, Hidden Markov Models. The basic idea was to combine a dictionary with a Hidden Markov Model to perform a 'soft-matching'[3] of phrases in the text to entries in the dictionary.

Kou et al. constructed a protein name dictionary by extracting the 'protein name' field from the PIR-NREF database[4]. They integrated the entries in the

[3]An approach that uses a text-similarity measure such as string edit-distance or vector-space cosine similarity to flexibly match textual items.

[4]http://pir.georgetown.edu/pirwww/search/pirnref.shtml

dictionary into a Hidden Markov Model, so that each sequence of states in the Hidden Markov Model corresponded to a single entry from the dictionary. The text was first stripped of stop words, and then tokenized. Then the tokenized text was passed through the dictionary-integrated HMM and resulting sequences of feature vectors were classified.

This method was evaluated on three datasets: University of Texas[5], GENIA[6], and YAPEX[7]. They achieved precision values of 73.4%, 49.2%, and 67.8% respectively; recall values were 47.8%, 66.4% and 66.4% respectively.

2.2.1.4 Zhou et al. (2004)

Zhou et al. [52] described a named entity recognizer, *PowerBioNE*, which also adopted a statistical machine-learning technique, a Hidden Markov Model. Their system dealt with various special characteristics of naming conventions in biomedical domains, including descriptive naming conventions, conjunction and disjunction, unknown words, abbreviations, and cascaded constructions. They claimed their system was the first system to deal with cascaded entity names.

In order to deal with these special characteristics, Zhou et al. incorporated into a HMM-based named entity recognizer a comprehensive set of evidential word formation patterns (i.e., capitalization, digitalization, etc.), morphological patterns (prefix, suffix), part-of-speech patterns, head noun triggers (the major noun of a noun phrase), special verb triggers (i.e., *bind*, *inhibit*, *activate*), and name alias features (i.e., abbreviations, short forms, etc.). The idea was to assign each output an appropriate tag, which contained boundary and class information, so that the sequences with the most likely tags would be extracted according to their likelihoods.

Zhou et al. evaluated the PowerBioNE system on GENIA V3.0 and GENIA V1.1. For GENIA V1.1, they selected 590 abstracts as the training data and 80 as the test data; for GENIA V3.0, they selected 1800 abstracts as the training data and 200 as the test data. Their results showed a precision of 66.5% and recall of 66.6% for GENIA V3.0 with a precision of 63.1% and recall of 61.2% for GENIA V1.1.

[5]ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/ proteins.tar.gz
[6]http://www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/posintro.html
[7]http://www.sics.se/humle/projects/prothalt

2.2.1.5 Mika et al. (2004)

Mika et al. [35] described a named entity recognizing system, *NLProt*, which combined a pre-processing dictionary-based and rule-based filtering step with several separately trained Support Vector Machines to identify protein names in MEDLINE abstracts. Figure 2.5 shows the architecture of NLProt.

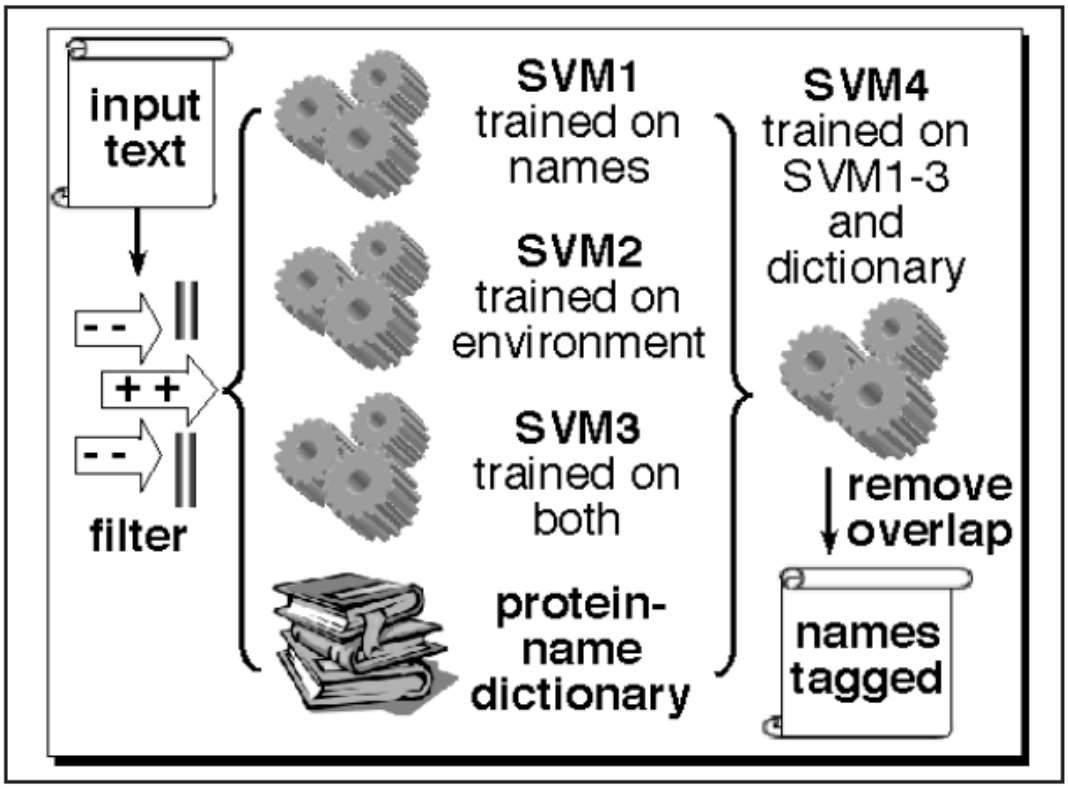


Figure 2.2: The architecture of NLProt [35]

First, the input text was 'sliced' into individual samples by a 'sliding window' approach. Secondly, these contiguous samples of tokenized words were pre-filtered through a prepared dictionary. The dictionary was a list of all SWISS-PROT + TrEMBL [7] protein/gene names with each name linked to its associated database identifier. Thirdly, the resulting words were passed to the three SVMs: SVM1, SVM2, and SVM3. Each of the SVMs was trained on a specific part of the samples:

SVM1 focused on the centers (i.e., names), SVM2 on the contexts surrounding the names, and SVM3 on the overlap between the two. Fourthly, the output values from the three SVMs were combined with a dictionary, and in turn generated the final score for each sample.

Mika et al. evaluated their system on GENIA. They selected 180 abstracts as the training data for the first three SVMs, 15 as the training data for the fourth SVM, and five as the test set. They then rotated through all abstracts, such that each abstract was used for testing exactly once. Their result showed that the use of both SVMs and a dictionary achieved an average precision of 75%, and recall of 76%.

2.2.1.6 Summary

It is difficult to compare the performance and efficiency of various approaches because of the use of different training and test data. In general, each approach has its respective advantages and disadvantages.

Dictionary-based approaches have several advantages over the other approaches. They can make use of the huge amount of information in curated databases; they require no training, therefore can perform more uniformly over different data sets; they also can be easily followed by the entity normalization process as they often provide identifiers for recognized words. However, dictionary-based approaches have two fundamental problems. The first is a large number of unknown words and false positives caused mainly by name variations, which typically results in low recall compared to the other approaches. The second problem arises because the extractors often become outdated when the dictionaries upon which they are based on become outdated.

Manually constructed rule-based systems demonstrate reasonable performance on biomedical texts because many entity names have recognizable word-format patterns like capitalization, digitalization, etc. The advantages of the rule-based methods include an ability to use the volumes of information in curated databases, the lack of need for less descriptive models, the lack of need for domain knowledge, and the ready extension using linguistic knowledge due to conceptually obvious rules. However, a disadvantage of rule-based systems is that rules need to be manually constructed and maintained. Another disadvantage is that these systems do not provide identification information on recognized terms, which simplifies the entity recognition process, but on the other hand is a serious drawback to the entity normalization process.

The context-based approaches have been shown to have the best performance among all three types of approaches, but they are often the most complex to implement. These approaches often require extensive training whereas the training collections for gene and protein normalizations are few. In principle, they do not require updating when the set of entities changes. However, machine-learning–based extractors do depend on the specified dictionary of entities in the test data being available at training time. It is unclear how they would perform on different test sets, hence they may not be readily applicable to different data.

2.2.2 Named Entity Normalization

A natural follow-up task to Entity Name Recognition is Entity Name Normalization [16]. Named Entity Recognition and Normalization are the fundamental tasks in biomedical text mining. Gene and protein named-entity recognition and normalization are often treated as a two-step process. While the first step, NER, has received considerable attention over the past few years, normalization has received much less. A typical NER system usually uses a combination of hand-built dictionaries, approximate string-matching, and parameter tuning based on the training data. The most common obstacles that a NER system might face are: name variations, synonyms, acronyms, and so forth. Below we describe three systems ([16], [51], and [40]), which each tackles one of the problems above using its own novel approach.

2.2.2.1 Cohen (2005)

Cohen [16] developed a dictionary-based NER system that required no training or manually built dictionaries. An integrated list of terms from several online database was constructed by extracting the standard symbol, unique identifiers, name, synonyms, and aliases from each of these databases. The input text was segmented into sentences, and each segment was then searched for the terms in the integrated dictionary. Cohen selected a total of five online databases: MGI[8], Saccharomyces[9], UniProt (the curated SwissProt portion only)[10], LocusLink[11], and the Entrez Gene database[12].

[8] http://www.informatics.jax.org
[9] http://www.yeastgenome.org
[10] http://www.pir.uniprot.org
[11] http://www.ncbi.nlm.nih.gov/LocusLink
[12] http://www.ncbi.nlm.nih.gov/entrez

To increase the performance of the system, Cohen adopted the following strategies:

Re-processing database:

1. Generate orthographic variants for each term in the list by iteratively applying a few simple rules, such as replacing internal spaces in the original term with hyphens (or vice versa), removing hyphens or spaces in the original term, etc.
2. Remove the 300 or so most common English words (stop words) from the dictionary.
3. Separate the dictionary into two parts, one part containing the terms easily confused with common English words (the 'confuse dictionary'), and a much larger dictionary of terms that are not likely to be confused with English words (the main dictionary). Terms in the confuse dictionary will be searched in the input text without regard to case while the main dictionary are case-sensitive.

Disambiguation:

Cohen proposed a unique disambiguation algorithm that was based on the assumption that usually either an author provides sufficient context for the reader to resolve ambiguous terms, or the ambiguous terms are synonyms for other non-ambiguous terms within the same text context.

Cohen evaluated the performance of the system on the BioCreative [1] Task1B mouse and yeast collections, and compared his results to other participants in BioCreative Task 1B. His system's precision was among the best, with an F-measure near the median. Cohen further evaluated the performance of each approach described above, and, surprisingly, the case-sensitive search for main dictionary terms produced the largest improvement in the F-measure, 15.6%, while removing stop words made the second largest improvement, 6.8%.

Cohen's results demonstrate that an simple, easily implemented and unsupervised dictionary-based approach to NEN can be as effective as more sophisticated systems. His experiments also give interesting insights into how various factors can affect the performance of dictionary-based approaches.

2.2.2.2 Pustejovsky et al. (2001)

A more complicated approach to recognizing entity names is *acronym-meaning identification*. Acronyms are widely used in the biomedical literature and other scientific texts, therefore the ability to recognize and link acronyms to their full-length reference terms is very important in improving the performance of biomedical information extraction. Pustejovsky et al. [40] presented a *Vector Space Model* algorithm for disambiguating the acronyms that have multiple meanings, *Polyfind*.

Polyfind specifically targets one case of ambiguity: A *polynym* (an acronym or alias that has several possible associated long forms or meanings) is found in a text, and the meaning is not available or defined in that text. The algorithm has two steps:

1. For the target polynym, build the training data by collecting MEDLINE abstracts that define the polynym for each of its meanings. Abstracts that define the same meaning are grouped together and will be used as document templates.

2. For a text that contains the polynym without definition, the similarity is computed between this text and the sets of document templates, each of which defines a meaning for the polynym. The polynym is assigned the meaning defined in the document template that has the highest similarity score.

Pustejovsky et al. evaluated the correctness of their algorithm against manual results: their algorithm achieved 97.62% accuracy in disambiguating acronym *SRF* and 82.22% in disambiguating alias *p21*. These results showed that a vector space model for polynym sense disambiguation is both applicable and effective to alias disambiguation.

2.2.2.3 Yu et al. (2003)

Yu et al. [51] investigated four existing approaches for extracting entity synonyms, with the prerequisite that the entity names must already have been identified in the text. The approaches included unsupervised, partially supervised, and supervised machine-learning methods, as well as manual knowledge-based methods. Yu et al. also developed a combined system that exploited the strengths of three of these techniques, and performed a thorough evaluation of five approaches.

An unsupervised approach: Contextual similarity.

This method determines whether a set of words are synonyms by comparing their contexts. If the contexts of the two words are similar, then the two words are considered synonyms. This method is based on the simple observation that synonyms of a word tend to appear in the same contexts.

A partially-supervised approach: Snowball.

Snowball uses a bootstrapping approach for extracting structured relations from natural language. In this case, the relation of interest is *Synonym(term1, term2)*. The system starts with a small set of user-provided positive and negative examples, then proceeds to recognize and generate patterns iteratively by searching for occurrences of positive pairs. The patterns are later used to extract synonyms. An example pattern could be "term1 also known as term2".

A Supervised approach: Text classifier SVM.

The classifier SVM starts with the same positive and negative examples used by Snowball. Then the classifier is trained to distinguish between the 'positive' text contexts and the 'negative' text context. After the classifier is trained, the system examines every context surrounding pairs of terms and determines whether the context is a positive or negative instance. Each pair of terms is then assigned a confidence score.

A manual knowledge-based approach: GPE.

Here, the patterns in which synonyms appear are generated manually by domain experts. These patterns are then used to automatically scan the text for additional new synonyms.

The combined system.

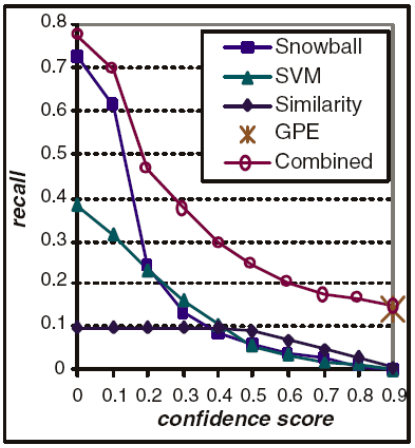
Each of the above approaches has its advantages and disadvantages. As an unsupervised method, Similarity does not require manual training, but it does not distinguish false positives from true positives. Snowball and SVM can extract patterns automatically, so are therefore able to determine more synonyms than the labour-intensive GPE system. However, GPE generates a smaller but higher quality set of synonyms as it is the least likely to extract false positives.

The combined system integrates the output of Snowball, SVM and GPE. The input text is processed by each system and pairs of synonyms are output, together with corresponding 'confidence' scores that represent the probability that the extracted synonym pair is correct. The combined system then

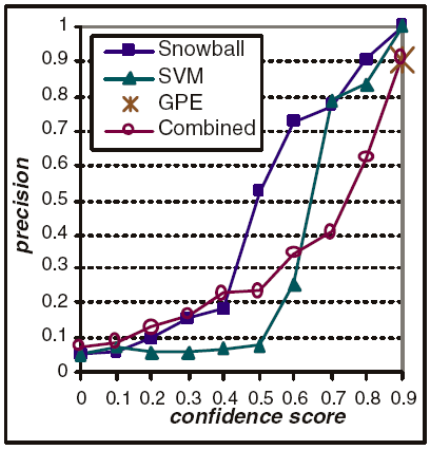
computes the final probability that the synonym pair was extracted correctly as (1 - the probability that all systems extracted this pair incorrectly).

Yu et al. evaluated Similarity, Snowball, SVM, GPE, and Combined over a collection of 52,000 recent biomedical journal articles collected by the GeneWays[13] project. Figure 2.3 shows the performance metrics of all systems. In summary, the combined system has the best performance for both precision and recall. Among all the systems, Similarity had the worst recall (less than 0.09% for all confidence score), and precision (less than 0.01%, too low to be shown). Snowball, SVM, and the combined system had comparable performances: the combined system had the highest recall for all confidence score, while Snowball and SVM outperformed the combined system on precision for all confidence scores larger than 0.6. As GPE always assigns the confidence score of 1 to all extracted candidate pairs, therefore GPE's performance was represented by a single data point in each plot. In terms of the estimated number of real synonym pairs extracted, the combined system had the largest estimated number. In summary, Combined was the best performing system for all metrics.

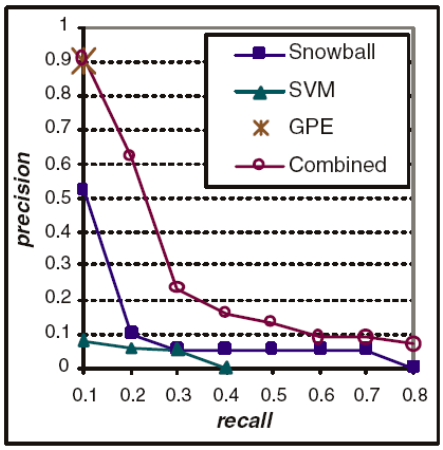
[13]http://geneways.genomecenter.columbia.edu/



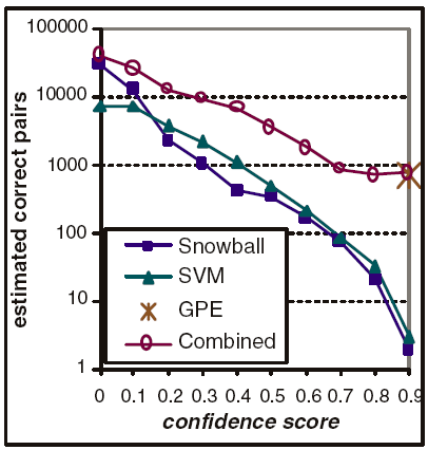
(a) Recall versus confidence score



(b) Precision versus confidence score



(c) Precision versus Recall



(d) Estimated number of synonym pairs correctly extracted by each system versus confidence score

Figure 2.3: Precision and Recall for Similarity, Snowball, SVM, GPE, and Combined [51]

2.3 Functional Annotation

Functional Annotation in biomedical domains can be described as the attachment of biological information to named entities identified in texts, e.g., proteins, genes, etc. An example of a basic annotation activity would include using BLAST [5] to determine sequence similarity and then annotating the genome on the basis of these similarities. Currently, an increasing amount of information is being added to the scope of annotation. This additional information, such as biochemical function, biological function, and participated interactions, could greatly help researchers in various genome studies. The need for accurate and robust automatic annotation tools has risen significantly due to the past decade's explosive increase in the numbers of genes and proteins both discovered and predicted. However, the Functional Annotation task is more difficult and complex than either Named Entity Recognition or Normalization because of the innate complexity of molecular functions, relations between molecular functions and molecular products, and the expressions of said functions and relations.

This section has provided a brief overview of current state of research studies on general Functional Annotation in biomedical domains. Functional annotation specific to Protein-Protein Interaction (PPI) will be described in the following section.

2.3.1 Relationship Extraction

There are many applications of Functional Annotation in the biomedical domain, with one of the most popular topics in recent years being the extraction of specific relationships between genes. The goal of this type of task is to detect occurrences of a certain relationship between a pair of genes or proteins. In brief, this task has been approached using three different types of methods: pattern-based, statistical, and Computational Linguistic. Pattern (template)-based methods use pre-existing templates to extract concepts linked by a specific relation from a text corpus. The templates are either manually generated by domain experts or automatically generated by detecting patterns in the context surrounding concept pairs known to share the relevant relationship. Statistical methods are used to identify relationships by searching for co-occurrence of concepts in the same text context, i.e., two proteins appearing in a sentence might indicate there is a relationship between these two proteins. These methods are based on the reasoning that concepts which co-occur more frequently than predicted by chance are likely to have some type of semantic relationship. Lastly, Computational Linguistic methods use knowledge-based

techniques to perform a substantial amount of text parsing to decompose the text into a representation from which relationships can be readily extracted. ([44], [47], [39]).

Sekimizu et al. [44] collected the most frequently used verbs in a collection of abstracts, then developed shallow-parsing techniques to find corresponding subjects and objects. They estimated their precision at 73%. Stapley and Benoit [47] extracted the co-occurrences of gene names from MEDLINE articles, then used this information to predict the relations between genes using their co-occurrence statistics. Pustejovsky and Castaño [39] targeted the extraction of *inhibit* relations from text and finite-state automata to recognize these relations. Their unique approach was based on the use of coreference relationships to extract *inhibit* relations that spanned multiple sentences. We will describe each of these three approaches in more detail in the following subsections.

2.3.1.1 Stapley and Benoit (2000)

Stapley and Benoit's [47] approach was based on the premise that if two genes have a related biological function, then there should be an increased likelihood of those two gene names occurring in the same document or document abstract. They investigated this hypothesis by generating graphs in which nodes represented genes and edges were representing the co-occurrence relation of two genes. Their experiment successfully showed linking of related genes, but no actual precision or recall rates were given. It is clear that this approach cannot extract more detailed information so it is of limited use. A great deal of useful information could be extracted from MEDLINE articles if the text could be more thoroughly analyzed to obtain useful linguistic knowledge such as morphological, syntactic, semantic, and discourse information.

2.3.1.2 Sekimizu et al. (1998)

In a more-sophisticated approach, Sekimizu et al. [44]'s aim was to recognize and extract from free text more-detailed information about interactions between proteins and other molecules by using templates that matched specific linguistic patterns of usage. Sekimizu et al. adopted a rather straightforward strategy to identify interactions between genes and gene products. Their method was centred on the frequently occurring verbs in MEDLINE abstracts. They identified interactions between genes by finding the subject and object terms for the most frequently occurring verbs in the raw text of the MEDLINE abstracts. In parsing each sentence

in this corpus, instead of traditional full parsing techniques they used partial and shallow-parsing methods to obtain morphological and syntactic information. Their method was able to find a large quantity of subject and object pairs for various verbs with a reasonable precision rate. The precision of those pairs was examined manually for various verbs, and depending upon the specific verb, the precision ranged from 67.8% to 83.3%.

Sekimizu et al. then attempted to automatically generate database entries containing relations extracted from MEDLINE abstracts. The relations in which they were interested were based on the following verbs: *activate*, *bind*, *interact*, *regulate*, *encode*, *signal* and *function*. This task formed part of a larger project which included automatic SGML tagging of abstracts before information extraction is performed. Their overall methodology was to parse, determine noun phrases, identify the commonly occurring verbs, then select the most likely subject and object from the candidate noun phrases in the surrounding text. They used a corpus of 898,000 words extracted from MEDLINE and reported precision results which ranged from 67.8% to 83.3% across the different verbs.

2.3.1.3 Pustejovsky and Castaño (2002)

Pustejovsky and Castaño [39] developed a robust parser for identifying and extracting *inhibit* relations from biomedical text. Their approach used a combination of lexical-semantic theory and large-corpus analysis techniques. They first constructed simple semantic-analysis automata for the relevant relations, then developed rules specific to a particular relation or a class of relations by doing a corpus analysis for the subset of MEDLINE abstracts corresponding to the target relations, e.g., *inhibit*. A distinguishing feature of their system was its anaphora resolution module. This module focused on the resolution of anaphoric dependencies within biomedical literature, (i.e., MEDLINE), and could be used to integrate entity identification and coreference resolution modules for information extraction in biomedical domains. The results reported in this paper focused on the extraction of *inhibit* relations and demonstrated that it was possible to extract limited, but biologically important, information from free text with high reliability using a classical natural language processing approach.

2.3.1.4 Summary

Although the template/pattern approach produces better results than the statistical approaches, it is still inherently limited: this type of system often targets only

abstracts, deals with only a single sentence at a time, and uses simplified methods of linguistic analysis. As a consequence, these approaches to biomedical information extraction miss a great deal of the detailed information on gene relations that are contained in the text. Potentially a great deal of additional information could be extracted from scientific articles if we were able to analyze the entire text of the article to derive detailed linguistic information such as lexical meanings, syntactic structure, semantic content, and discourse structure.

Computational Linguistics research is still not sufficiently advanced to handle these difficult problems even for restricted sub-languages and certainly not for the very large corpora needed for useful biomedical information extraction. Various systems have attempted to finesse these difficulties by using a method of text analysis that approximates full syntactic processing, and that takes a heuristic approach to semantic analysis based on the recognition of interactions between proteins and other molecules in the form of templates matching specific linguistic patterns. However, current research results in relation extraction indicates that significant improvement is still needed to make the existing systems effective in practical applications.

2.3.2 Protein-Protein Interaction Extraction

Many applications have now emerged that target a broad range of extraction problems in protein-protein interaction (PPI). Representative approaches to extracting protein-protein information from biomedical texts include: simple template-based parsing of sentences to build networks of protein interactions [12]; a general-purpose information-extraction engine using both symbolic and statistical Computational Linguistic techniques to build a database of protein interactions [48]; using the frequency of 'discriminating words' to score paper abstracts to determine whether the paper is about protein interactions [34]; and assessment of the reliability of protein-protein interaction using an 'interaction generality' measure [43].

2.3.2.1 Blaschke et al. (1999)

Blaschke et al. [12] attempted to do without linguistic analysis such as parsing, and relied instead on a simple pattern-matching approach for extracting protein interactions from MEDLINE texts. The text was first broken into clauses, then clauses containing two proteins and an action verb were extracted with simple syntactic ordering information used to predict the relation. For example:

'*protein1 action protein2*': makes *protein1* the subject, *protein2* the object and *action* the relation.

The 'interaction' verbs used included *acetylate*, *activate*, *destabilise*, *inhibit*, *phosphorylate*, *suppress* and *target*. The task was simplified by assuming that all protein names were already known and by not attempting to produce any quantitative assessment. The basic idea was that sentences derived from abstracts will contain a significant number of protein names 'connected' by verbs that indicate the type of relation between them. The method was based on counting the number of sentences containing protein names separated by interaction verbs. By prespecifying a limited number of possible verbs, Blaschke et al. avoided the need for complex semantic analysis.

The system design relied heavily on the peculiarities of the subject domain. The test corpus of abstracts contained a very specialized type of texts, including a very restricted use of English, short sentences, and a great abundance of highly specialized terms in Molecular Biology. This inflexibility inevitably led to missed relations and false negatives. For example, this system would be unable to deal with cases in which a subject or object was at a distance from a verb, e.g., parentheticals, relative clauses, and so on. However, with an adequate number of abstracts in the test corpus, the system could be subjected to a quantitative analysis, in which the number of occurrences of different events were more significant than the single occurrence of a valid event.

2.3.2.2 Thomas et al. (2000)

Thomas et al. [48] modified an existing information extraction system, *Highlight*, for the task of gathering data on protein interactions from MEDLINE abstracts. Highlight is a template-based system for general information extraction used by commercial applications. Thomas et al. modified Highlight in the following steps to make it feasible for biomedical information extraction.

1. Add new vocabulary, technical terms, and syntactic constructs in protein domains to the system so that the customized system can correctly recognize the relevant entities and events.

2. Construct the templates that outline the biomedical information of interest.

3. Construct patterns that represent the events of interest in text. A set of pattern-matching rules and statistical components are also developed. These

rules will be used for inserting the identified entities and events into corresponding templates.

Thomas et al. initially manually analyzed about 200 abstracts to discover the most frequently used verbs to describe protein interactions. In the end they decided to use only three key verbs *interact with*, *associate with*, *bind to*, as these verbs all appear in direct relations between proteins rather than between a protein and some process. They collected 2565 abstracts from MEDLINE as test data by searching for keywords *molecular*, *interaction* and *protein* for the year 1998. They then evaluated the customized Highlight System by extracting protein-protein interactions from the test data. The recall and precision values were estimated by taking three samples of 30 abstracts each and analyzing them by hand. The precision was 69% and the recall was 29%.

2.3.2.3 Marcotte et al. (2001)

A frequently encountered problem in protein-protein interaction extraction is the lack of a high-quality training corpus consisting of 'pure' protein-protein interaction articles. Marcotte et al. [34] proposed a method to mine literature describing protein-protein interactions by computing word occurrence frequencies. The words that appeared at unexpectedly high or low frequencies were identified as *discriminating words*. Marcotte et al. collected the MEDLINE abstracts of 260 papers that were cited by the Database of Protein Interactions [50]. These abstracts were used as the true positive training data. Then, for each word in the training data, its word frequencies were computed in both general biomedical texts and protein-protein–interaction texts. In the end, a total of 84 discriminating words were determined, among which the most discriminating words included terms such as 'Complex', 'Interaction', 'Two-Hybrid', 'Interact', 'Proteins', 'Domain', etc. Marcotte et al. then used a Bayesian approach to scan the target MEDLINE abstracts, and each abstract was given a 'likelihood' score for its probability of discussing the topic of Protein-Protein Interaction according to the discriminating words observed in the abstract.

The authors tested their method on 325 Yeast *MEDLINE* abstracts and compared their results with manual classification. They found that more than 88% interaction abstracts had a positive likelihood score and any articles that had a likelihood score higher than 10 had a 100% chance of belonging to the protein-protein interaction domain.

2.3.2.4 Saito et al. (2003)

Although there has been a great deal of research on protein-protein interaction extraction and identification, few attempts have been made to validate the interactions either recorded in interaction databases or newly extracted from literature. Experimental data on protein interactions often contains many false positives. Saito et al. [43] proposed a method that measured the reliability of interactions by observing the other proteins that interact with the target pair. This measure,*interaction generality*, was based on the hypothesis that an interaction occurring in the literature was likely to be false positive if the interacting proteins appeared to have many other interacting partners, but those partners had no further interactions. In contrast, highly interconnected sets of interactions or interactions forming a closed loop were likely to be true positives.

This 'interaction generality' measure incorporated the topological properties of interactions around the target interacting pairs. As an illustration, Figure 2.4 shows the classification of a protein that interacts with the target interacting pair according to the topological properties of the interaction network. A and B are the target pair; C is the third protein that interacts with either A or B or both; the black-filled circle represents another protein that interacts with C . Interactions in class *a1*, *a2*, and l all form a closed loop, thus correspond to true positive. In contrast, the proteins in class d and f are 'weakly' connected, thus the target interaction is likely to be false positive.

Saito et al. tested their method on the relatively reliable protein interaction data sets, and the experiment results showed that there indeed existed relations between the 'interaction generality' measure and the reliability of an interaction.

2.3.3 Summary

In this chapter, we have given an overview of the current state of information extraction in biomedical domains. In the following chapters, we will propose a discourse-analysis–based approach to extracting information from protein-protein interaction literature and will investigate the hypothesis that this information can be used to evaluate the biological validity of protein-protein interaction. The method uses both sophisticated Computational Linguistic methods and computationally tractable algorithms capable of processing large corpora.

We base our hypothesis on the inherent biological characteristic of protein-protein relationships, namely, that interacting proteins will tend to have similar biological functions. We may reasonably expect then to find biological terms in

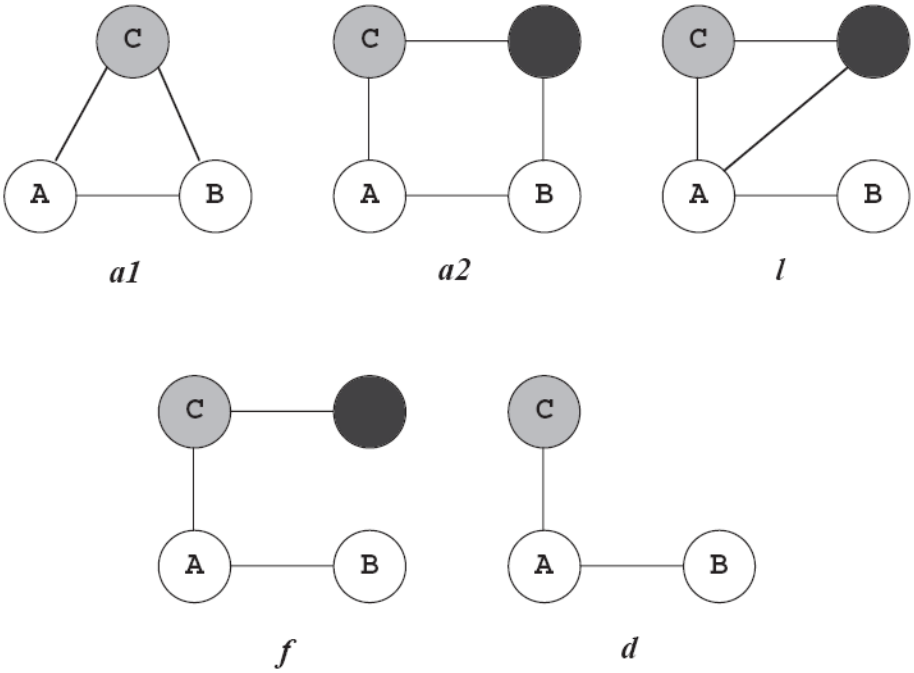


Figure 2.4: Classification of a protein C that interacts with a target interacting protein pair A-B [43]

the context surrounding a protein interaction that indicate the common functions of these proteins. If we can determine such terms by an automated method of linguistic analysis, we would have an additional means of discovering evidence in the literature that the interaction is indeed biologically valid.

The idea of using semantically related strings of words to determine the discourse structure of text is known as *lexical chaining* [36], a method that fulfills our dual criteria of being both discourse-based and computationally efficient. We propose to use lexical chains to retrieve additional information on protein interactions by finding the biological terms in the passage surrounding an interaction that form the theme structure of the text. By constructing the lexical chains related to protein interactions, we will not only extract additional important information about interactions from the literature, but we hypothesize that we will also be able to use the strength of the chains as a basis to rank the apparent quality of the interactions.

Chapter 3

Lexical Chaining

In this chapter, we will describe the basic concepts of Lexical Chaining techniques, and then present the lexical chaining algorithm we adopted for our experiments.

3.1 Lexical Cohesion

The notion of *Lexical Chaining* derives from the concept of textual cohesion. Halliday and Hasan [26] referred to *cohesion* as "relations of meaning that exist within the text, and that define it as a text". The linguistic study of textual cohesion shows that a text or discourse is not just a set of sentences, each about some random topic. Rather, the sentences and phrases of any sensible text tend to 'stick together' by various means to form a unified whole. There are a number of forms of textual cohesion, such as grammatical cohesion (reference, substitution, ellipsis, conjunction) and lexical cohesion (i.e., semantically related words).

Lexical Cohesion arises from semantic relationships between words, and is the most frequent and most easily identifiable type of cohesion. In linguistics, lexical cohesion is used to explain one aspect of how a text's meaning is created, through "continuity of lexical meaning" [26]. Halliday and Hasan [26] classified lexical cohesion into two categories, *reiteration* and *collocation*. Reiteration includes not only repetition and reference, but also superordinates, subordinates, synonyms[1], hypernyms[2], and hyponyms[3]. Collocation was defined as a semantic relationship

[1]**synonym**: a word that means the same as another word, or more or less the same

[2]**hypernym, superordinate**: a general term that includes various different words representing narrower categories, called **hyponyms**

[3]**hyponym, subordinate**: a word that is more specific than a given word

between words that often co-occur in the same lexical contexts. Halliday further defined three basic classes for reiteration, and two basic classes for collocation, as shown in Figure 3.1.

Examples 1 and 2 represent the simplest form of reiteration: repetition; example 3 represents the *superordinate* relation: *peach* is a kind of *fruit*; example 4 demonstrates the *antonymy* relation: *green* and *red* are members of an unordered set {*white, black, red*, etc.}, which falls into the category of *systematic semantic collocation*; *garden* and *digging* in the example 5 have a *non-systematic semantic* relationship.

3.2 Defining a Lexical Chain

Lexical cohesion occurs only between two terms, but may lead to sequences of related words. A *lexical chain* may then be defined as a sequence of semantically related words in the text, spanning a topical unit of the text, be it short (adjacent words or sentences) or long (entire text). As an illustration, the following passage has a sequence of related words.

1. John has a **Jaguar**.
2. He loves the **car**.
3. John works in the **garage** taking care of his **Jaguar**.

In this passage, the word *Jaguar* in sentence 1 and sentence 3 is a repetition; *Jaguar* and *car* has a IS-A relationship; *car* and *garage* form a collocation that is not systematically classifiable. A lexical chain would therefore be: {*Jaguar*, *car*, *garage*, *Jaguar*}.

In general, each document will contain many lexical chains, each of which forms a portion of the cohesive structure of the document. Lexical chains are important for computational text understanding, because they not only provide a context for resolving word ambiguity, but also indicate the discourse structure of the text. Morris and Hirst [36] were the first researchers to use lexical chains to determine the structure of texts. Their results showed that the lexical chains retrieved from a text will tend to mirror the discourse structure of that text.

- **Reiteration with identity of reference:**
  Example 1

  1. Mary bit into a *peach*.
  2. Unfortunately the *peach* wasn't ripe.

- **Reiteration without identity of reference:**
  Example 2

  1. Mary ate some *peaches*.
  2. She likes *peaches* very much.

- **Reiteration by means of a superordinate:**
  Example 3

  1. Mary ate a *peach*.
  2. She likes *fruit*.

- **Systematic semantic relation (systematically classifiable):**
  Example 4

  1. Mary likes *green* apples.
  2. She does not like *red* ones.

- **Non-systematic semantic relation (not systematically classifiable):**
  Example 5

  1. Mary spent three hours in the *garden* yesterday.
  2. She was *digging* potatoes.

Figure 3.1: Definitions and examples of Lexical Cohesion [36]

Since then, lexical chaining has been successfully used in a number of Information Retrieval and Natural Language Processing applications, such as term weighting [46], malapropism detection [28], hypertext generation [25], and text summarization [9]. Hirst and St.-Onge [28] used manually constructed lexical chains for the detection and correction of malapropisms. Stairmand [46] used lexical chaining in the construction of both a typical Information Retrieval system and a text segmentation system [46]. Green [25] developed a technique to automatically generate hypertext links using lexical chaining. Barzilay and Elhadad [9] used lexical chains to weight the contribution of a sentence to the main topic of a document, and sentences with larger weight are extracted and presented as a summary of that document.

As a lexical chain "encapsulates" a context, most lexical-chaining-based text summarizer follows the observation that the "strength" of the lexical chain corresponds to the semantic significance of the textual context it represents. We hypothesize that the strength and other characteristics of lexical chains can be used as a basis for the assessment of the biological validity of protein-protein interactions. In our experiments, we used the lexical chains to extract information from Protein-Protein-Interaction (PPI)–related literature, then we investigated a set of measurements for lexical chains, such as strength, lemma, density[4], etc.

3.3 A Lexical Chaining Algorithm

Lexical Chaining is the process of extracting and connecting semantically related words from a text, then creating a set of word chains that represent the various topic "threads" through the text. Generally speaking, lexical chains can be computed by grouping sets of words that are semantically related (words that have relationships such as identity, synonymy, and hypernymy/hyponynmy). In terms of actual computing procedures, most lexical-chaining algorithms can be summarized by the following three steps:

1. Select a set of candidate words (i.e., all noun instances).
2. For each candidate word, find an appropriate chain relying on a relatedness criterion among members of the chains.
3. If such a chain is found, insert the word in the chain; otherwise a new chain is created.

[4]The metrics will be discussed in Chapter 5

The difficult, and computationally costly, part of this process is that each candidate word must be assigned to exactly one lexical chain, and the words must be grouped in such an optimal way that these groupings create the longest/strongest lexical chains. There are several feasible algorithms for constructing lexical chains from a text, from which we chose Silber and McCoy's [45] algorithm for its simplicity and linear runtime. Silber and McCoy's algorithm was based on the complete method implemented by Barzilay and Elhadad [9], which runs in exponential time, but Silber and McCoy managed to obtain a linear running time with similar output. In constructing lexical chains, Silber and McCoy used WordNet, an online lexical database (to be discussed below) as the knowledge source for the lexical semantic relationships. The algorithm is shown in Figure 3.2.

In our experiment, we used the adaptation of Silber and McCoy's lexical-chaining algorithm implementated by Matthew Enss [20]. Enss' implementation differs from Silber and McCoy's algorithm in three aspects:

1. Silber and McCoy's algorithm disambiguates words during the computing of the lexical chains, specifically in Steps 2.1 and 2.2. In Enss' implementation, word disambiguation is separated from computing lexical chains and is performed by using the most accurate method available. Enss argued that improved word sense disambiguation necessarily leads to improved lexical chains. His experiments showed a significant decrease in the number of incorrect lexical chains generated when using the second approach.

2. When choosing a metachain for a candidate word, Silber and McCoy's algorithm considers only the most closely related word. It uses the strongest relation between the candidate and the other words in the metachain as the contribution of the candidate word to that metachain. Enss computes the contribution of a term to a metachain by summing the scores between the term and every other tems in the metachain. Enss' experiments showed that this modification increases the accuracy of word sense disambiguation from 42.9% to 52.1% when tested on the same corpus. However, there is a drawback, namely, the runtime of the algorithm is increased to $O(n^2)$, where n is the number of candidate terms in the document.

3. When there is a tie among metachains, Silber and McCoy's algorithm chooses the chain with the more specific overriding senses. Enss' algorithm favours the chain with more general overriding senses to allow for larger chains, which he believed are more representative of the overall subject of the text.

3.4 WordNet: A Linguistic Knowledge Resource

Lexical chains are built by using linguistic resources that relate words by their meanings. The original work by Morris and Hirst [36] used *Roget's Thesaurus* [14], but almost all current lexical chainers use the WordNet database ([21], [3]). *WordNet* is an online linguistic database that was created by a team of linguists and psycholinguists at Princeton University. Distinguished from a traditional lexical dictionary, WordNet aims to be a combination of dictionary and thesaurus that is useful for real-world Computational Linguistics and natural language processing applications. To achieve this goal, WordNet models the lexicon based on psycholinguistic principles so that words are grouped into sets of synonyms, *synsets*, according to word senses. Synsets are the core meaning units of WordNet, and a synset is a set of all the words or collocations that are synonyms for a particular sense of a word.

WordNet synsets may be linked to one another by a number of lexical semantic relations. The complete WordNet database contains four components, each for a different type of English word: nouns, verbs, adjectives, and adverbs. WordNet distinguishes between the four types of words because they follow different grammar rules, consequently the lexical semantic relations vary between the types of word. Nouns are the core part of WordNet with their possible relations defined as follows:

hypernymy : Y is a hypernym of X if every X is a (kind of) Y

hyponymy : Y is a hyponym of X if every Y is a (kind of) X

coordinate terms : Y is a coordinate term of X if X and Y share a hypernym

holonymy : Y is a holonym of X if X is a part of Y

meronymy : Y is a meronym of X if Y is a part of X

Each WordNet sense has three attributes: (1) a synset that lists all the synonyms for the sense; (2) a unique integer identifier for the sense; and (3) a gloss that describes the sense. If a word has multiple meanings, then it appears in multiple synsets. Table 3.1 shows all the noun senses of *car* in WordNet 2.1.

Synset	Index	Glossary	Direct Hypernym
{car, auto, automobile, machine, motorcar}	02929975	a motor vehicle with four wheels; usually propelled by an internal combustion engine	{motor vehicle, automotive vehicle}
{car, railcar, railway car, railroad car}	02931574	a wheeled vehicle adapted to the rails of railroad	{wheeled vehicle}
{cable car, car}	02906118	a conveyance for passengers or freight on a cable railway	{compartment}
{car, gondola}	02932115	the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant	{compartment}
{car, elevator car}	02931966	where passengers ride up and down	{compartment}

Table 3.1: All noun senses of *car* in WordNet 2.1

WordNet is the current standard lexical database for English and is widely used in most lexical-chaining algorithms. However, our research on extracting biological information from protein-protein-interaction–related literature requires a domain-specific ontology that has the same structure as WordNet, but different domain-specific vocabulary. We therefore propose to build an ontology that incorporates into the existing *Gene Ontology (GO)*[5] a selection of concepts representing protein-protein interaction domain-specific knowledge, together with the semantic relationships among these concepts. The new ontology will then be constructed in the form of a WordNet-like lexical database.

In the next chapter, we will describe in detail the protocol we developed to construct our "PPIWordNet" lexical database that includes Gene Ontology's current vocabulary and a set of other biological terms relevant to protein-protein interaction research.

[5]The Gene Ontology will be discussed in the next chapter

3.5 A Definition of Semantic Relatedness

In Silber and McCoy's algorithm, a critical component involves determining the relatedness of words making up a lexical chain. Initially, a noun is put into a metachain if it is in some way related to the sense with which the metachain is indexed. Subsequently, the degree to which the word contributes to the metachain must be measured in order to decide which metachains will be kept. In order to do this, we need a means of measuring the semantic relatedness of words.

There are various WordNet-based semantic relatedness measurements (e.g., [29], [28], [8]). Intuitively, the senses connected by a path in WordNet should also be related to some degree, due to semantic relations such as hypernymy and hyponymy being transitive. Moreover, the shorter the path, the more closely the senses would appear to be related. However, one problem with using hypernymy/hyponymy paths to determine relatedness is deciding the maximum length of a path to allow.

Hirst and St.-Onge's method proposed a simple but efficient way to measure the semantic relatedness using only the hypernymy and hyponymy relations in WordNet. Their idea was that two lexicalized concepts are semantically close if their WordNet synsets are connected by a path that is not too long and that "does not change direction too often". They adapted Morris and Hirst's semantic distance algorithm, which used Roget's thesaurus, for use with WordNet. Their method views semantic relationships between words in terms of a graph, and correlates semantic relatedness between words with the nature of the corresponding path between concepts in the graph. Semantic relatedness is then determined based on the path shape and distance between concepts using the relations connecting them in the WordNet taxonomy.

Hirst and St.-Onge [28]'s measure classified WordNet relations as having direction (upward, downward, or horizontal), and then established the relatedness between two concepts A and B by finding a path that was neither too long nor that changes direction too often. Three kinds of relations were defined: *extra-strong* (between a word and its repetition), *strong* (between two words connected by a WordNet relation), and *medium-strong* (when the link between the synsets of the words is longer than one and satisfies certain restrictions). As an example, two words are strongly related if one of the following holds:

1. They are members of the same synset (e.g., *human* and *person*).
2. They are associated with two different synsets connected by the antonymy relation (e.g., *human* and *object*).

	Same sentence	Within 3 sentences	Same paragraph	Default
same synset	1	1	1	1
parent or child	1	0.5	0.5	0.5
sibling	1	0.3	0.2	0

Table 3.2: Silber and McCoy's term-based score function [20]

3. One of the words is a compound (or a phrase) that includes the other, and there is any kind of link at all between the synsets associated with each word (e.g., *school* and *private school*).

Two words are said to be in a medium-strong relation if there exists an 'allowable' path connecting the synsets associated with each word. An allowable path involves certain patterns of links between synsets that may vary among upward (hypernymy and meronymy), downward (hyponymy and holonymy), and horizontal (antonymy).

In Hirst and St.-Onge's scheme, the strength of a lexical chain is based both on its length and the types of relationships among its members. Extra-strong relations have the highest weight, next in weight are strong relations, and lowest are medium-strong relations. Unlike extra-strong and strong relations, medium-strong relations have varied weights according to the following formula ([28], p. 308):

weight = C – *path length* – $k *$ *number of changes of direction* (where C and k are constants[6].)

We originally adopted Hirst and St.-Onge's measure in our manual study of lexical chains. However, this measure is hard to implement and computationally costly. Enss' implementation used the same measure used by Silber and McCoy. This measure considers both the factor of semantic distance between word senses and the locations of the terms in the text. The function is shown in Table 3.2. More discussion on the semantic relatedness measure functions will be given in later chapters.

[6] C has value 8 and k has value 1 (Graeme Hirst, personal communication).

- **Preprocessing:**

  1. Tokenize input text
  2. Tag each token with appropriate part-of-speech tagger.

- **Step 1: Creating the metachains**

  1.1 Implicitly build all possible 'metachains' for each sense of a word in WordNet; a single metachain represents all possible lexical chains for that core meaning.

  1.2 *for each noun in the document*
  for every sense of the noun in WordNet
  Place the noun sense into every metachain for which it is related to that sense. For two senses to be considered related, they must be either the same sense (in the same synset), a parent-child pair, or siblings (children of the same parent).

- **Step 2: Computing the best chain**
for each word in the document
for each chain that the word belongs to

  2.1 Find the single metachain for each noun that the noun contributes to most. The contribution of a candidate word to a metachain is measured by the strongest relation between the candidate word and the other words in that metachain. The semantic relation between two words are calculated based on the type of relation and distance factors. For example, identity and synonymy are considered equally strong contributors to a lexical chain over a passage of three sentences, but hypernymy is considered less strong over the same distance.

  2.2 When there is a tie among metachains, choose the chain with the more specific overriding senses.

  2.3 Remove the candidate word from all other metachains to which the word belongs. When this step completes, each noun will belong to only one metachain. When all the nouns have been processed, the optimal lexical chains will remain.

Figure 3.2: Silber and McCoy's lexical chaining algorithm

Chapter 4

A Protocol for Constructing a Domain-Specific Ontology: PPIWordNet

4.1 Motivation

The main goal of our research is to extract information that can be used to evaluate the biological validity of protein-protein interactions using lexical-chaining techniques. We can reasonably assume that the targetted information stored in chains should consist of PPI domain-specific terms/concepts. In the Artificial Intelligence field, as Neches et al. [38] put it, an ontology "defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary". Our research requires a domain-specific ontology that covers important concepts and their relationships for the Protein-Protein Interaction domain, and that can also be easily 'plugged into' our lexical-chaining analysis module.

4.1.1 The Gene Ontology

Initially, we chose the Gene Ontology as our domain-specific ontology. Among various biological ontologies, the *Gene Ontology* (GO) is presently the most widely used ontology. The Gene Ontology project is a collaborative effort to address the need for consistent and precise descriptions of genes and gene products in any organism. The project began as a collaboration between three model organism

databases: FlyBase (Drosophila)[1], the Saccharomyces Genome Database (SGD)[2], and the Mouse Genome Database (MGD)[3].

The Gene Ontology consists of three structured sub-ontologies: biological processes, cellular components, and molecular functions. A biological process refers to a biological objective to which the gene or gene product contributes. A process is accomplished via one or more ordered assemblies of molecular functions. A molecular function is defined as the biochemical activity (including specific binding to ligands or structures) of a gene product. A cellular component refers to the place in the cell where a gene product is active.

Each term in the Gene Ontology has a unique identifier, name, definition, and any synonyms, and is assigned to one of the three sub-ontologies. The Gene Ontology also defines two relations between terms, *IS-A*, and *PART-OF*. IS-A is a simple class-subclass relationship, where "A IS-A B" means that A is a subclass of B; for example, *nuclear chromosome* IS-A *chromosome*. PART-OF is slightly more complex: "C PART-OF D" means that whenever C is present, it is always a part of D, but C does not always have to be present. For example, nuclei are always part of a cell, but not all cells have nuclei, thus the relations between these terms can be described as "*nucleus* PART-OF *cell* nuclei".

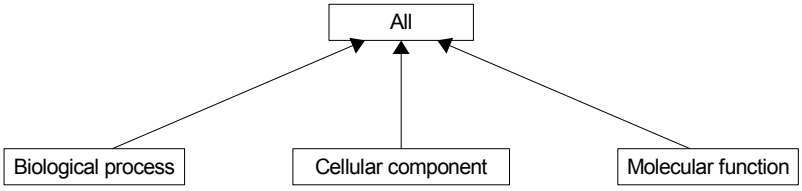
Terms in the Gene Ontology are linked by these two relations, and linked terms form a directed acyclic graph where a parent may have multiple children and a child may have multiple parents, but no parent-child relation loop is allowed. The structures and examples of the high-level terms in each sub-ontology are shown in Figure 4.1. Obviously, the Gene Ontology's structure is very similar to that of WordNet, which makes the Gene Ontology a suitable lexical knowledge resource that can be easily restructured into a lexical database in the same format of WordNet and used by our application.

As of 2006, the Gene Ontology has grown into the most widely used biological ontology, containing over 17,000 terms from many databases, including several of the world's major repositories for plant, animal, and microbial genomes. However, not all biological domains are covered by the Gene Ontology. The protein-protein–interaction domain is one of the domains that the Gene Ontology has so far chosen not to include. Without a vocabulary describing various aspects of protein-protein

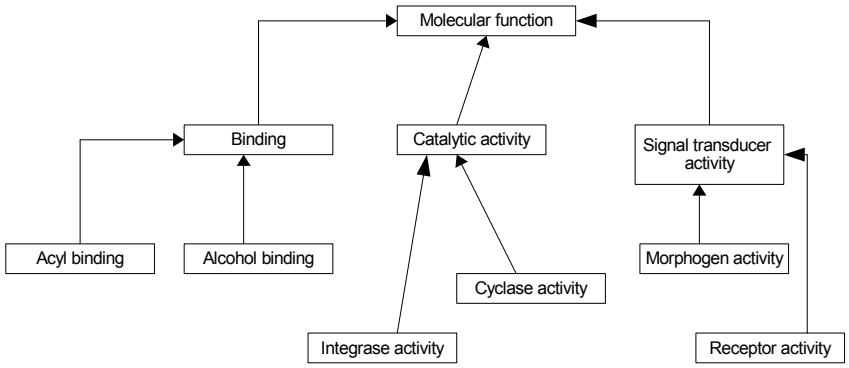
[1]FlyBase is a database of genetic and molecular data for Drosophila. http://flybase.bio.indiana.edu/

[2]SGD is a database of the molecular biology and genetics of the yeast Saccharomyces cerevisiae. http://www.yeastgenome.org/

[3]MGD is one component of the Mouse Genome Informatics (MGI) system (http://www.informatics.jax.org), a community database resource for the laboratory mouse.

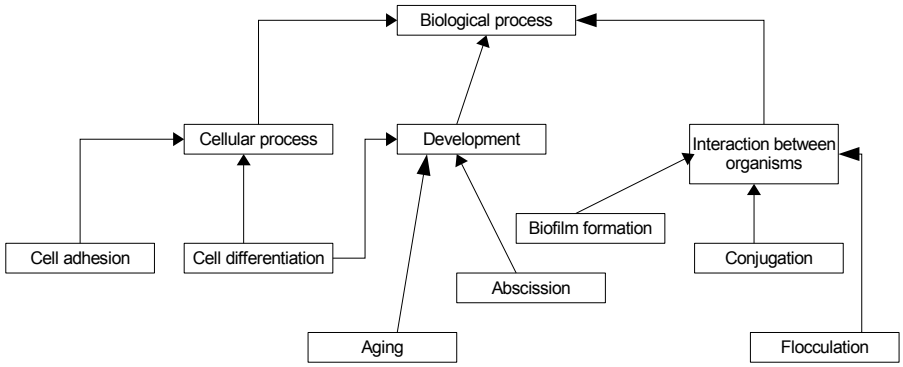


(a) The sub-ontologies of the Gene Ontology

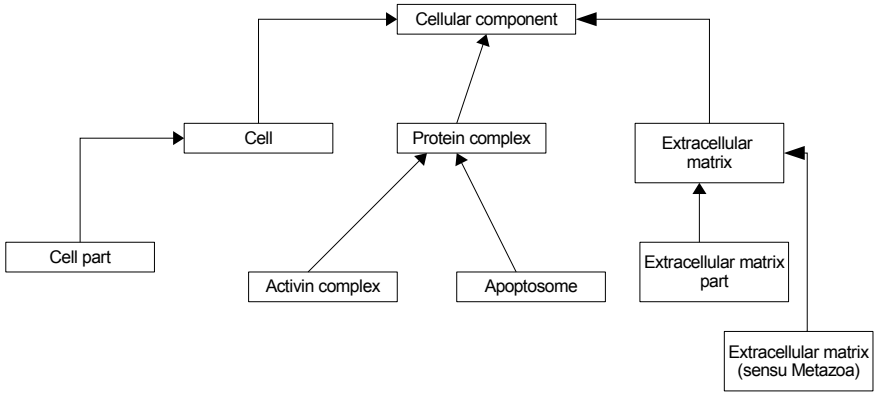


(b) Molecular Function

Figure 4.1: Structure and examples of the Gene Ontology



(a) Biological Process



(b) Cellular Component

Figure 4.2: Structure and examples of the Gene Ontology (Continued)

interactions, the lexical chains constructed by using GO alone will not be sufficient for our goal. Our initial experiments using the Gene Ontology alone has shown this. It was evident that it would be necessary to construct a PPI domain-specific ontology to suit our needs. Our expectation in developing this ontology is to create a lexical knowledge resource to use in the extraction of information about protein-protein interactions. This newly constructed PPI domain-specific ontology will then be integrated into the Gene Ontology as arguments. Our proposed "PPIWordNet" lexical database will include GO's current vocabulary and a set of other biological terms relevant to protein-protein interaction research.

This chapter is organized as follows. Section 4.2 gives a short overview of related work in ontology construction and our hybrid approach. Details of our methodology and the information retrieval techniques we used are presented in Section 4.3, followed by a summary in Section 4.4.

4.2 Overview: A Hybrid Approach

4.2.1 Related work on ontology construction

Ontology construction is the construction of a conceptually concise basis for communicating, sharing, and reusing knowledge in specific domains over different applications. In general, ontology construction is a difficult and complex process. The two main challenges in ontology construction are: ontology concept capture (how the concepts in a domain can be discovered) and relationship determination (how the relationships between concepts are determined). Different approaches have been used for building ontologies, but most can be classified into one of the two main categories: manual methods and automated methods.

An example of a manual method is the one proposed by Uschold and King [49]. Their method involved three stages: identifying the purpose of the ontology (i.e., why to build it, how it would be used, range of the users), building the ontology, evaluation and documentation. The building of the ontology was further divided into three steps:

1. **Ontology Capture.** Key concepts and relationships are identified, precise textual definitions of these are written, terms to be used to refer to the concepts and relations are identified, and the actors involved agree on the definitions and terms.

2. **Ontology Coding.** The ontology is coded in a formal language to represent the defined conceptualization.

3. **Ontology Integration.** The ontology is integrated with other existing ontologies if possible.

A manual method of ontology construction typically involves the initial manual extraction of commonsense knowledge from various sources, and the subsequent manual construction of relationships among these knowledge concepts. Although the manual process guarantees the quality of the concept structures to some extent, there are two common drawbacks: low coverage and high expense. For example, domain experts must be involved in defining the boundaries of the ontology and they must determine the terms for the defined boundaries of the ontology, based on their own expertise and a variety of relevant sources, such as indexes, encyclopedias, handbooks, textbooks, journal articles, as well as any existing and relevant thesauri or vocabulary systems. It is extremely difficult and time-consuming for human experts to construct an ontology from given data or texts. Usually only a very limited number of the most important top-level concepts can be covered. Moreover, as the correctness and quality of an ontology is directly linked to the experts' knowledge about the targetted domain, it is hard for manual methods to avoid inconsistencies in the ontology. One solution to the problems is to build an ontology automatically or at least, using semi-automatic methods. In addition, after an ontology is built, it must be constantly updated to reflect the change of information within the area, which often cannot be done thoroughly, given the slow response times of the manual process.

Automated methods often rely on corpus-based statistical approaches that automatically build domain-specific concept structures. Recently, a number of workshops at Artificial Intelligence and Machine Learning conferences (ECAI[4], IJCAI[5], ECML/PKDD[6]) have been organized on learning ontologies. Several papers on automated ontology development by concept extraction and learning have been presented at these conferences, including extending the existing WordNet ontology using Web documents [4], using clustering for semi-automatic construction of ontologies from parsed text corpora [11], learning taxonomic, e.g., *IS-A* [15] and non-taxonomic, e.g., *HAS-PART* [33] relations.

[4]ECAI is the biennial European Conference on Artificial Intelligence. http://ecai2006.itc.it/

[5]IJCAI is the International Joint Conference on Artificial Intelligence. http://www.ijcai.org/

[6]ECML is the European Conference on Machine Learning and the European Conference on Principles and Practice of Knowledge Discovery in Databases, http://www.ecmlpkdd2006.org/

Typically, an automatic approach first selects suitable text corpora to represent the domains of interest, then finds statistical evidence about terms in the text corpora, then finally determines relationships between concepts by considering statistical evidence in text corpora. The key issues in automated approaches are how to determine and qualify different kinds of statistical evidence for an optimal relationship determination. The most commonly used form of statistical evidence is the co-occurrence frequencies of terms in the texts. Recent approaches have begun to take into account the actual semantic content in the context of a co-occurrence. In summary, the research so far in automated ontology construction indicates that significant improvement is still needed to make the existing approaches more effective in practical applications.

4.2.2 Our hybrid approach

The goal of this chapter is to present a hybrid approach for the construction of a domain-specific ontology, *PPIWordNet*, for the protein-protein interaction domain. This approach combines the semi-automatic extraction of domain concepts and the manual construction of relationships between concepts. In the process of concept extraction, concepts are extracted from source texts and added into the ontology according to two factors: 'discriminality' and 'importance'. 'Discriminality' indicates how discriminating a term is for a certain domain compared to other knowledge domains, and 'importance' represents how important/relevant a term is to the biological evaluation of protein-protein interactions.

The *specificity* of a term is calculated by comparing the word frequencies of the term in a corpus of full-text protein-protein interaction articles to a general-English text corpus using a statistical information retrieval method. The *importance* of a term is manually determined by domain experts based on their knowledge about the PPI domain. By combining importance and specificity, we are able to weight and sort terms in a text corpus, and choose a limited number of top-ranked terms as concepts for the PPI domain ontology. In the process of relationship construction, we use two types of relations: *IS-A* and *ASSOCIATED-WITH*. Relationship determination between concepts is performed manually and collaboratively by knowledge engineers and domain experts using these two relationships. For the validation of our approach, two domain experts manually validate the results every step of the way.

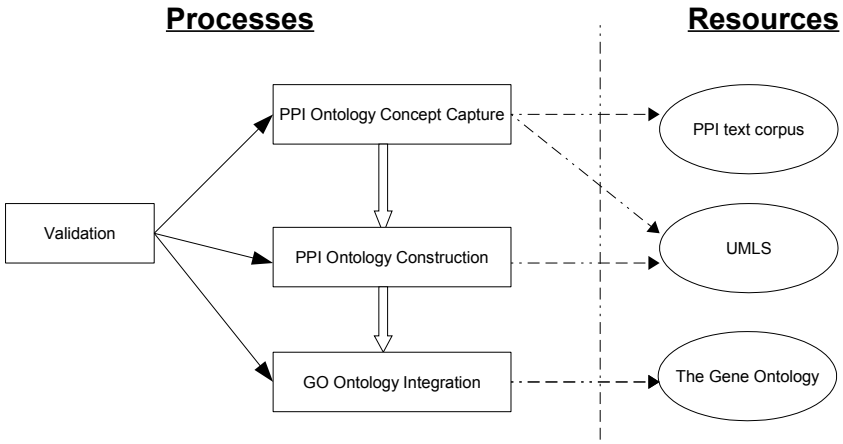


Figure 4.3: A module for the construction of PPIWordNet

4.3 Development Methodology

The proposed module for developing the PPIWordNet ontology is shown in Figure 4.3. It consists of four processes (PPI Ontology Concept Capture, PPI Ontology Construction, GO Ontology Integration and Validation) and three data resources (PPI text corpus[7], the Gene Ontology, and the Unified Medical Library System (UMLS) [2]. Each stage of the process will be discussed in detail in the following subsections.

4.3.1 Process 1: PPI Ontology Capture

The goals of this process can be summarized as follows:

1. To capture and identify key concepts and relationships in the PPI domain;
2. To give precise and unambiguous natural language definitions for such concepts and relationships;

[7]The text corpus used will be discussed in detail later on.

3. To identify the terms that represent/refer to such concepts and the synonyms of those terms.

This process is broken down further into four steps:

4.3.1.1 Step 1: Source selection

Choosing an appropriate source of text data as the basis from which concepts will be extracted is the first step in corpus-based concept extraction. A good domain-specific source should have sufficient domain coverage, contain as little noise as possible, and be readily available. For all these reasons, we choose the training data provided for the BioCreative PPI evaluation tasks[8]. The training data was derived from the content of the IntAct[9] and MINT[10] databases. The training data contains a total of 1,000 articles, with all articles manually reviewed to verify the content described protein interactions. These articles were used to manually extract the protein interactions mentioned, linking each interacting protein to its corresponding unique UniProt ID (or accession number) and providing the identifier of the described interaction detection method. These articles are capable of providing considerably more domain information than traditional text corpora, therefore highly suitable for our goals.

For a text corpus representing a large and diverse collection of general-English text, we choose the latest copy of the Wikipedia[11]. The Wikipedia is the largest multilingual free-content encyclopedia on the Internet. Containing over two million articles and still growing, the Wikipedia is considered by some researchers to be more more representative of general English than any of the standard NLP and TREC newspaper collections, etc.

4.3.1.2 Step 2: Extraction of Discriminating Terms for the PPI domain

Discriminating terms are terms that appear at statistically significantly higher frequencies in texts about a certain domain than in general text corpora. Our assumption is that the over-represented terms are likely to represent the key content of a text, thus in turn being most relevant to the specific domain. Every term in the

[8] http://biocreative.sourceforge.net/biocreative_2_ppi.html
[9] http://www.ebi.ac.uk/intact/
[10] http://cbm.bio.uniroma2.it/mint/
[11] http://en.wikipedia.org/wiki/Main_Page

domain-specific text corpus is examined and weighted, and then a set of discriminating terms will be selected based on their weights. The goal of term weighting is to give a quantitative measure for computing the discriminatity of a term. We adopted an elaborated weighting method based on the frequency of occurrence, *tf-idf*, where the weight of a term i in a document j is determined by the frequency of the term i in the document j and the number of documents containing term i in the text collection.

In the extraction of the PPI discriminating terms, we have been assisted by a doctoral researcher with a background in Information Retrieval methods[12]. The detailed process of discriminating-terms extraction is shown below:

Prepare the PPI articles for text analysis:

The format of the PPI articles was not in plain-text format. Thus, the text articles needed to be extracted from the PPI HTML articles. First, JTidy was used to convert HTML to XHTML. Then, an XML parser was used to parse the XHTML files to extract the plain text.

Remove the stop words from the PPI articles:

A list of stop words, words that have no significance in the PPI articles, such as *a*, *an*, *the*), were removed from the PPI text articles.

Term weighting:

For each PPI article, the term frequency/inverse document frequency (*tf-idf*) for each term in the articles was calculated using the following formula:

$$tfidf(t\_k, d\_j) = \#(t\_k, d\_j) * log(Tr/\#Tr(t\_k)) \tag{4.1}$$

where
$\#(t\_k, d\_j)$ denotes the number of times term t_k occurs in document d_j
$\#Tr(t\_k)$ denotes the number of documents in Tr in which t_k occurs
Tr denotes the number of documents

PPI term matrix:

For the corpus of PPI articles, a term matrix and a unique list of terms of the entire PPI articles were generated. Each row in the term matrix represents an article. Each column represents a term appearing in an article and the weight of the term. The frequency of each term in the unique list is calculated by

[12]We are indebted to Shady Hassan from the University of Waterloo's Computer Engineering department for advice and assistance concerning Information Retrieval.

averaging the frequencies of the same term in the PPI articles in which this term appears.

Prepare Wikipedia for text analysis:
An XML parser was used to extract only the Wikipedia full-text articles from the entire Wikipedia database.

Remove the stop words from the Wikipedia articles:
The same list of stop words that were removed from PPI articles were also removed from the Wikipedia text articles.

Wikipedia term matrix:
For the Wikipedia articles database, a term matrix and a unique list of terms of the entire Wikipedia articles were generated. Each row in the term matrix represents a Wikipedia article. Each column represents a term appearing in an article and the weight of the term. The frequency of each term in the unique list was calculated by averaging the frequencies of the same term in the Wikipedia articles in which this term appears. The generated list of unique terms in Wikipedia represents the most common English terms.

Prepare a list of the names of the protein families:
An XML parser was used to extract the names of the protein families along with their alternative names from the Human Protein Reference Database (HPRD)[13]

Prepare a list of compound names:
A list of compound names that contains inorganic compounds (compounds without a C-H bond), organic compounds (compounds with a C-H bond), and biomolecules was prepared.

Generating the discriminating terms:
The list of discriminating terms was generated by removing from the PPI unique list all the terms which appeared in the Wikipedia, protein, and compounds lists.

As an illustration, the 50 discriminating terms with the highest tf-idf value are shown in Table 4.1.

[13]The Human Protein Reference Database represents a centralized platform to visually depict and integrate information pertaining to domain architecture, post-translational modifications, interaction networks and disease association for each protein in the human proteome. http://www.hprd.org

Term	tf-idf Value	Term	tf-idf Value
pellino	0.9960094	topbp	0.66685784
pkc	0.98770636	bret	0.6657517
mirk	0.9760501	hmtase	0.66337526
atrap	0.94974506	dronc	0.6539748
chord	0.93665785	dronc	0.6539748
dysbindin	0.9159207	endosperm	0.6492104
alboaggregin	0.8851284	endosperm	0.6492104
daxxc	0.8815422	gabp	0.64609134
cambd	0.8790759	alix	0.6415269
rheb	0.87230194	wrnh	0.63805676
dicer	0.87055403	ergic	0.6243057
ikkq	0.81820816	toxcat	0.6190614
iiaglc	0.7984184	exosome	0.618216
srpk	0.7778112	pist	0.613866
smurf	0.77406406	iasys	0.60933936
scfmet	0.7707197	ckis	0.60915375
flipl	0.76781094	chimaerin	0.60130525
pygo	0.76378804	rabphilin	0.5917763
dokr	0.7610833	clpp	0.5902393
pmca	0.7575839	taci	0.58829147
allm	0.684885	mgrb	0.5833869
brap	0.680523	pbaf	0.57815033
midas	0.6734595	pstpip	0.5590316
asap	0.6701755	hsnf	0.5554925
dmyc	0.6679697	ribosyl	0.55532527

Table 4.1: The top-ranked 50 discriminating terms

The list of discriminating terms contain a total of 13,931 terms with a tf-idf value larger than 0. Upon a closer look at the list, we have two observations: first, even though our method has already filtered out protein names and chemical compounds using prepared lists, a large number of terms are of no interest to PPI research, including person names, gene names, chemical compounds, and proper nouns; secondly, the top-ranked terms are not necessarily all valuable for PPI research. As our experiment is a pilot project to examine whether our lexical-chaining approach to biomedical information extraction is effective, accuracy in constructing the ontology is more important than coverage for our application. The list of 13,931 terms were passed onto the domain experts for manual filtering. Two domain experts independently selected all terms that they considered relevant to PPI research. The two sets of terms were then merged together. A term was considered relevant and selected for the final list only when it appeared in both of the two sets. The resulting list of this step consisted of less than 200 terms relevant to PPI research.

4.3.1.3 Step 3: Build the glossary of discriminating terms

Information in WordNet is organized around logical groupings called *synsets*. Each synset consists of a list of synonymous words or collocations (a string of two or more words, connected by spaces or hyphens, which co-occur more often than would be expected by chance). Words or collocations are organized into hierarchies based on the hypernymy/hyponymy relation between synsets. In this step, the PPI domain experts built a glossary which included all the discriminating terms, their natural language descriptions, and their synonyms and near-synonyms. The natural language descriptions were not created following the style of traditional dictionaries, but rather by referring to other terms/concepts and including notions such as more abstract concepts (superclass), more specific concepts (subclass), etc. For example, concept *automobile* can be defined as the following:

***automobile*:** It is a kind of *vehicle*; It includes *cars* and smaller *suvs*, but not *motorcycles*, *buses*, *trucks* or *vans*;
Synonyms: *car*, *auto*, *machine*, *motorcar*.

If a term has multiple senses, each sense is defined as a unique sense in the same way described above. On the other hand, different terms that have the same sense will be merged into the same sense, and listed as synonyms of the sense. As an example, *automobile* has two senses:

1. As a noun, a motor vehicle with four wheels;

2. As a verb, travel in an automobile.

Thus *automobile* should be defined as two independent concepts, each with a different sense. While *motorcar* has only one sense and that sense is the same as the noun sense of *automobile*, *motorcar* will be added into *automobile*'s synonym set, instead of being defined as a unique concept.

The domain experts used the Unified Medical Language System (UMLS) [2] as the referencing resource in this task. UMLS has two knowledge resources that are of interest to us: the Metathesaurus, and the Semantic Network. The Metathesaurus is a very large, multi-purpose, and multilingual vocabulary database that contains information about biomedical and health-related concepts, their various names, and the relationships among them. The Semantic Network provides a consistent categorization of all concepts represented in the UMLS Metathesaurus and a set of useful relationships between these concepts. The domain experts used these two resources to find the natural language definitions and synonyms for the discriminating terms. Both resources were obtained from the UMLS website: http://www.nlm.nih.gov/research/umls/about_umls.html.

As this task was carried out by two domain experts, to maintain consistency, the glossary was built by one expert, and reviewed by the other expert[14].

4.3.1.4 Step 4: Identify the Seed Terms

The goal in this step was to identify the starting-point terms, from which small networks of interconnected terms could be constructed in Process 2 (PPI Ontology Construction). Each individual network constructed was then integrated into the PPIWordNet ontology in Process 3 (GO Ontology Integration).

There are three possible strategies in constructing a network of connected concepts: bottom-up, top-down, and middle-out. The bottom-up strategy identifies the most specific concepts first, and then generalizes them into more abstract concepts. The top-down strategy finds the most abstract concepts first, and then specializes them into more specific concepts. These two strategies both have their own advantages and disadvantages. The bottom-up strategy may result in a very high level of detail compared to the top-down strategy, but it can also lead to increased overall

[14]In the construction of the PPI ontology, we are indebted to Gabriel Musso and Zhou Yu, doctoral researchers in Biochemistry from the University of Toronto, for advice and assistance concerning PPI knowledge extraction.

effort and a risk of inconsistency. The top-down strategy has a better control of detail, but may result in arbitrary or unnecessary high-level categories, which could lead to instability in the model.

We decided to use the middle-out strategy, which first identifies a core of basic domain concepts, and then specifies and generalizes these as necessary. Compared to the other two strategies, the middle-out strategy strikes a balance in terms of level of detail. Detail is only included when necessary by specializing the basic concepts, while the high-level concepts created are more likely to be natural and stable as they arise from the most important and basic concepts.

To accommodate the middle-out strategy, in this step the domain experts identified the key domain terms and made them the seed terms. To do so, for each term in the list, the expert assigned a score by evaluating the degree to which the information might be considered important or useful when analyzing a protein-protein interaction. The score was based on a five-point scale: Very important/Useful, Important/Useful, Fair, Not Important/Useful, Don't Know, with Very Important/Useful being 5 , Important/Useful being 4, and so on. Note that a term could have multiple senses, and each of the senses could have different scores regarding protein-protein interaction relativeness. To simplify, we limited the number of senses for each term to two so that the domain experts would only take the two most relevant senses. On the other hand, different terms that had the same sense were treated as one term and its synonyms.

The Gene Ontology divides its terms into three categories: *biological process*, *cellular component*, and *molecular function*. We added two more categories: *interaction property* and *method*. *Interaction property* contains the terms that describe the attributes or properties of protein-protein interactions, while *method* contains the names of experiments or methods that are used to determine or confirm protein-protein interactions. In addition to the importance score, the domain experts also assigned each term to one or more categories.

In the end, a total of 54 terms labeled as Very Important/Useful were selected as the seed terms; the domain experts started building local ontologies around the seed terms in the next process. Figures 4.4 shows part of the list of seed terms selected.

4.3.2 Process 2: PPI Ontology Construction

This process is executed progressively. We started with the category molecular function, constructed a sub-ontology from all the terms belonging to this category,

Term	Definition	Synonyms	Categories
activatable	the property that can increase the rate of a biological process	accelerable	Molecular Function, Interaction Property
activator	a molecular (protein, coupound, etc.) that regulates one biological process by increasing the rate of reaction	accelerator, activating agent,catalyst, sensitizer,enhancer	Molecular Function, Interaction Property
activators	plurality of activator	accelerator, activating agent,catalyst, sensitizer,enhancer	Molecular Function, Interaction Property
affinity	attraction force between one substance (protein, resin, etc.) to others (protein, ligand, small moleculars)	attraction, attractive force, relationship, attractiveness	Method, Interaction Property
affinitypurified	purified using affinity approaches	pulldown, affinity column	Method
associate	a molecule (e.g., Protein) or biological event that always accompanies or closely relates to another	bind, accompany, connect, closely relate, inlvove,	Molecular Function, Interaction Property
associated	a molecule (e.g., Protein) or biological event that always accompanies or closely relates to another	accompanied, connected, closerelated, involved	Molecular Function, Interaction Property
binding	the capacity of connecting other molecules through physical interactions	connecting, holding, attracting, interacting, adhering, sticking, attaching, tie, associating	Molecular Function, Interaction Property
coactivate	activate together with something else	coaccelerate, coincrease	Molecular Function, Interaction Property
cocrystallized	a protein or small ligand (e.g., peptide and compound) that are crystallized with another protein in the process of crystallization, usually strongly indicating direct interaction	X-ray, crystallization, 3D-structure	Method

Figure 4.4: Examples of the seed terms

and then added one more category in each cycle. Each cycle can be divided into four steps:

4.3.2.1 Step 1a: Construct local hierarchy for each seed term

A *local hierarchy* that consisted of the directly related terms was created for each seed term. Here the relations of interest were IS-A, PART-OF, and ASSOCIATED-WITH. Even though only the IS-A relation is used in our present lexical-chaining analysis module, the other two relations could be valuable for future researches. The domain experts added parents, children, siblings for each seed term belonging to the currently 'working' category.

4.3.2.2 Step 1b: Expand the local hierarchy

This step was executed recursively. If the relevant terms added in the previous step appeared in the discriminating list and had a importance score of 4, the relative terms were expanded, and so forth, until no new terms with a score of 4 could be added.

4.3.2.3 Step 2: Link local hierarchies together

Local hierarchies were linked together if common terms were found between the hierarchies. The constructed network of connected local hierarchies can be displayed as a graph, with terms being nodes, and the relation being edges. It is natural that very high-level nodes in the ontology will have many paths through/to them, but it is not appropriate to include the entire subgraphs under those nodes. The ontology builder must determine which subgraphs should be included in the final ontology. The criterion taken into account in this task is that if many of the nodes in a subgraph have been found to be relevant to the PPI domain, then the other nodes in the subgraph are likely to be relevant too. IS-A is a transitive relation, which means if A IS-A B, and B IS-A C, then A IS-A C. However, our hierarchy does not allow loops. The Ontology Builder must restructure the network to avoid relation loops and optimize the structure of the network for our application.

An example of linked local hierarchies in the category *molecular function* is shown in Figure 4.5.

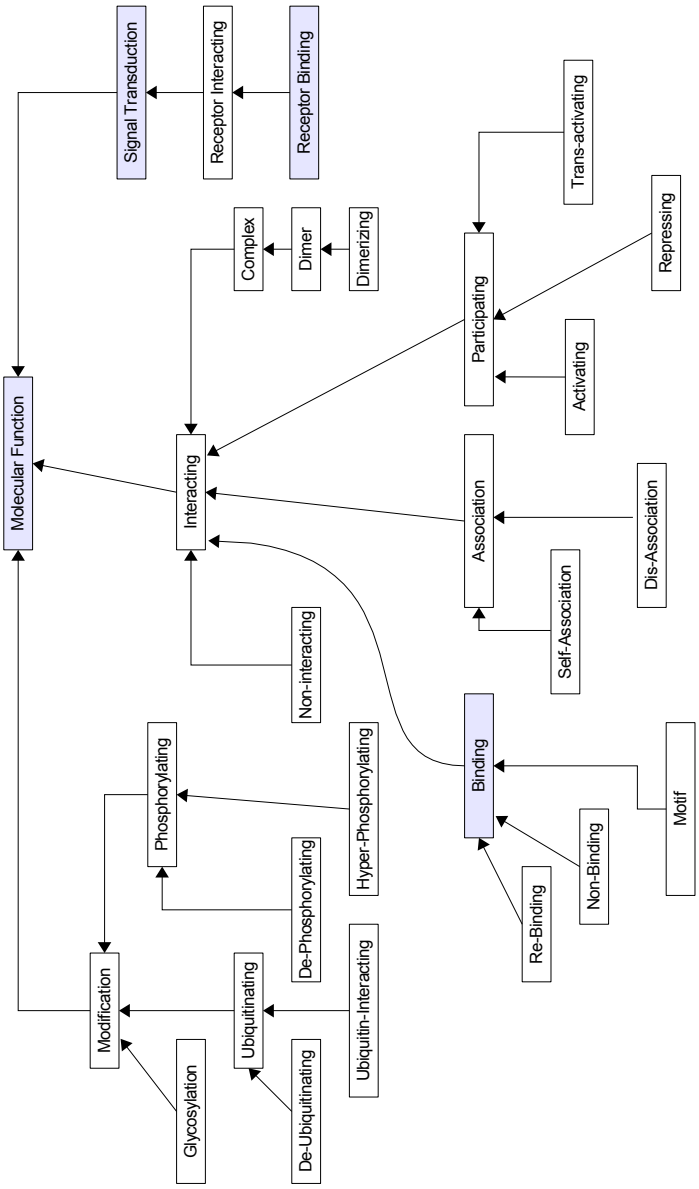


Figure 4.5: The sub-ontology for PPI molecular function terms

4.3.2.4 Step 3: Refine the ontology constructed.

The ontologies generated from the last step are further refined by structuring the terms more formally. Terms that are of different types of word, but can be represented by the same concept are formalized into one concept. An example of the refined ontology for molecular function terms is shown in Figure 4.6.

4.3.3 Process 3: Integrating with Gene Ontology

In the last step, our ontology of PPI domain terms was merged into the Gene Ontology. There are three cases of merging:

1. The same concept is identified in both the PPI ontology and the Gene Ontology, e.g., *binding* in Molecular Function. The concepts were merged together as one concept. Parents and children of the concepts from two ontologies were also merged. When there were contradictions between the two ontologies, the more-detailed one was favored. For example, in the PPI ontology, Binding IS-A Interaction, and Interaction IS-A Molecular Function, while in the Gene Ontology, Binding IS-A Molecular Function. In this example, the more-detailed relations in the PPI ontology would be adopted, as shown in Figure 4.7.

2. The same category is identified in both the PPI ontology and GO, e.g., Molecular function. As the GO developers created concepts for each of its categories, this case is dealt with the same way as the case above: the categories are merged together.

3. Unique categories in our PPI ontology, e.g., Interaction property. This type of category was added as an independent subset into the Gene Ontology.

4.4 Summary

In this chapter, we proposed a hybrid approach to construct a WordNet-like domain-specific ontology from a corpus of protein-protein interaction texts, and then integrate the resulting 'PPIWordNet' ontology into the Gene Ontology. Progressing through the complex processes of our method, we found the major difficulties were the conceptualization of domain terms and the formalization of relationships. Both

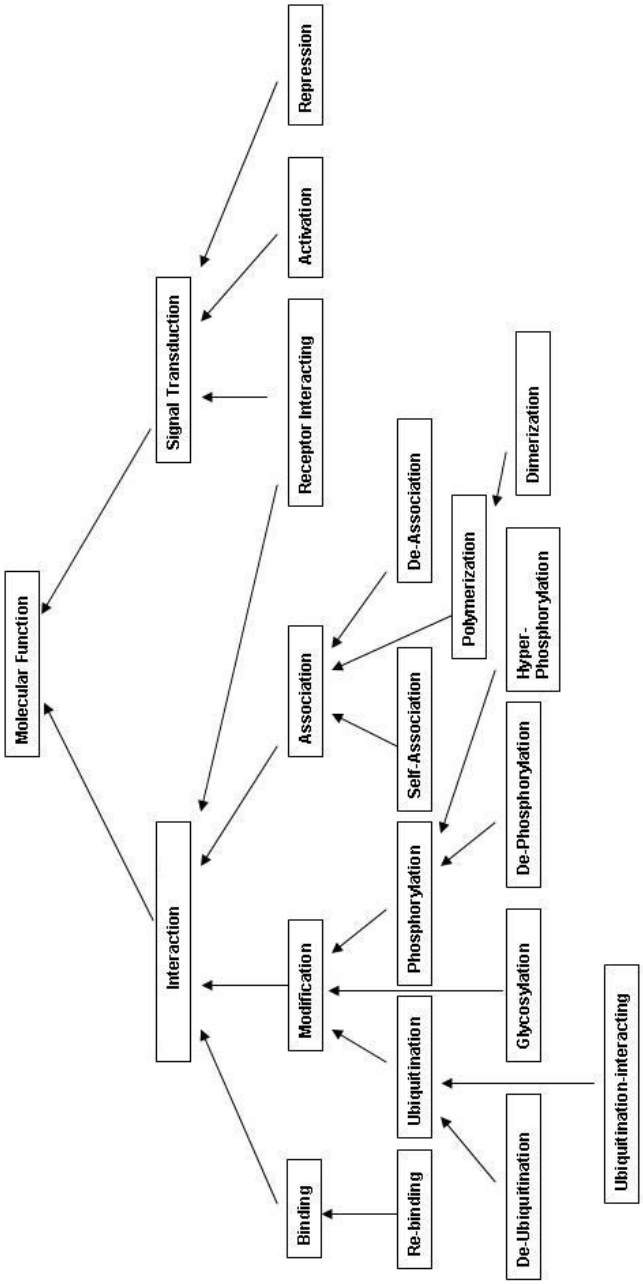


Figure 4.6: The final ontology for PPI molecular function terms

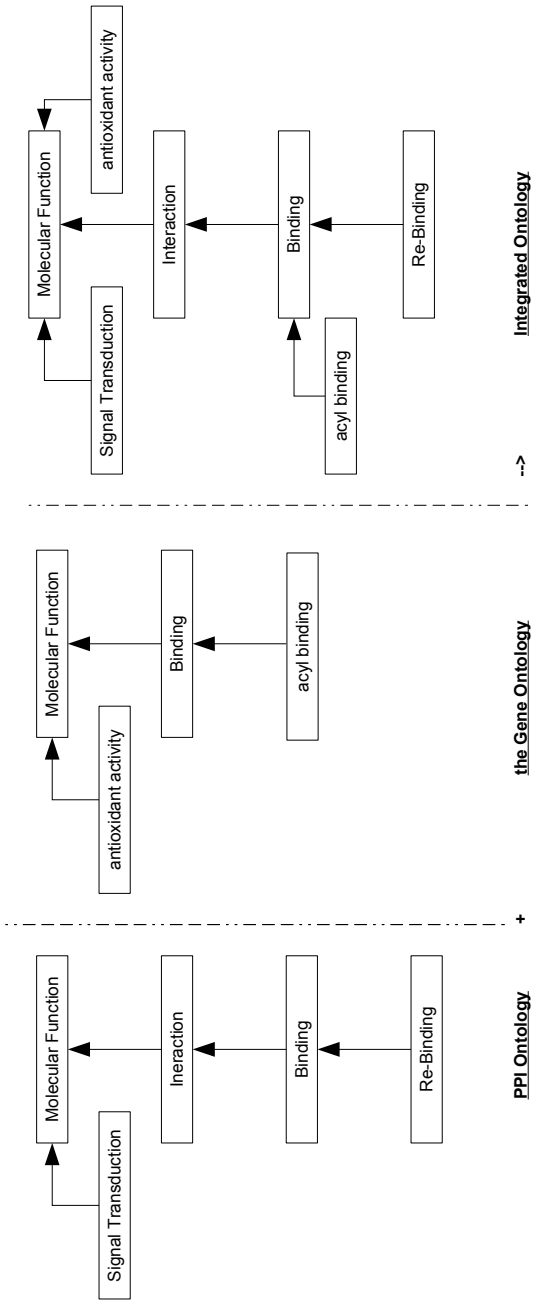


Figure 4.7: Integration with the Gene Ontology

activities relied heavily on the knowledge of domain experts about the protein-protein interaction domain. Eventually we would hope to develop an effective Ontology Construction Module that automates the process to some degree by using available online biomedical knowledge resources. Clearly further research and development is needed to automate our method.

The integrated ontology, PPIWordNet, was subsequently used by our lexical-chaining analysis module to extract information from PPI literature. The experiments and results will be described in detail in the next chapter.

Chapter 5

Experiment

5.1 Purpose of the Experiment

There are several distinct goals for this evaluation. First, we wish to investigate the discourse structure of protein-protein–interaction (PPI) texts using lexical-chaining analysis. Secondly, we wish to test our "PPIWordNet" ontology in a biomedical information extraction task, namely, lexical-chaining analysis of protein-protein interaction full-text articles to extract strings of semantically related words. Finally, our eventual goal is to use lexical chains as evidence of the biological validity of the protein-protein interactions in the contexts in which they appear. The first two goals have been achieved through our experiment, while progress has been made toward the final goal in future work.

5.1.1 Hypotheses of this experiment

The hypotheses to be tested in our evaluation are as follows:

1. Lexical chains appear throughout PPI articles and can be evaluated by various metrics relating to their quantity and quality.
2. Our PPIWordNet ontology is effective as a means of enabling discourse-based analysis of PPI texts using lexical chaining

5.2 Steps of the Experiment

In this section we experimentally evaluate our PPIWordNet ontology. To achieve our goals and test our hypotheses, we applied our lexical-chaining method and PPIWordNet ontology to a set of PPI-specific literature. We randomly selected 100 articles from the BioCreative PPI task's training data as the test set. These articles contain detailed information that has been used to identify protein-protein interactions. In choosing these articles, we hoped to find a good sampling of the kinds of biological terms likely to occur in protein-protein–interaction contexts, and which could ultimately be used to judge the quality of protein-protein interactions. The lexical-chaining algorithm we used was originally designed for extracting chains spanning over the whole text. We customized it so that it computes lexical chains on a paragraph-by-paragraph basis, as descriptions of protein interactions seldom spread across more than one paragraph as observed in our manual study. When a document is read in, it is first broken into paragraphs, then sentences, at last words. Then the lexical-chaining algorithm is performed on each paragraph. The total number of paragraphs in the test set was 2461.

To test the effect of our PPIWordNet ontology on lexical-chaining performance, we compared the chains created by progressively adding more terms from the PPIWordNet and then running the lexical-chaining algorithm. The PPIWordNet ontology was divided into three subsets based on its innate structure: Method, Interaction Property (IP), and Molecular Function (MF). In the first step, only the original Gene Ontology was used; in each step following, a subset of our PPIWordNet was added. In doing so, we hoped to find evidence that indicates our PPIWordNet has a positive impact on the extracted strings of semantically related words, in terms of both quality and quantity.

The steps of the experiment are shown in Figure 5.1. A statistical analysis on each set of results for the whole test set was performed and several measurements were computed. We then investigated the discourse structure of protein-protein–interaction texts by closely studying the lexical chains generated for a randomly selected article in the test set using the whole PPIWordNet.

- **Step 1**: Use only Gene Ontology (GO) on test set
- **Step 2**: Use GO + Method terms on test set
- **Step 3**: Use GO + Method terms + Interaction Property terms on test set
- **Step 4**: Use GO + Method terms + Interaction Property terms + Molecular Function terms on test set

Figure 5.1: Steps of the experiment

5.3 Results

5.3.1 Statistical analysis and performance metrics

We performed a statistical analysis on the lexical chains generated using different lexical sources. The test set has a total of 100 documents, 2461 paragraphs, and each chain spans over a paragraph. Before we present the details of the results, we introduce the metrics used in the statistical analysis. Note the metrics were computed for the whole test set.

# of terms: The total number of terms in the ontology used, synonyms not counted.

# of chains: The total number of chains generated.

# of chains per doc: The average number of chains generated per document.

# of chains per para: The average number of chains generated per paragraph.

Span: The maximum distance between terms in a chain, counted in sentences.

Lemmas: A lemma is the string indicating the base form of the term. For example, the lemma for *running* and *ran* is run. The lemma is used to find the possible senses of the word in the ontology. Lemmas indicate the number of unique lemmas in a chain.

Length: The number of terms in a chain.

Density: The number of terms divided by (span+1)

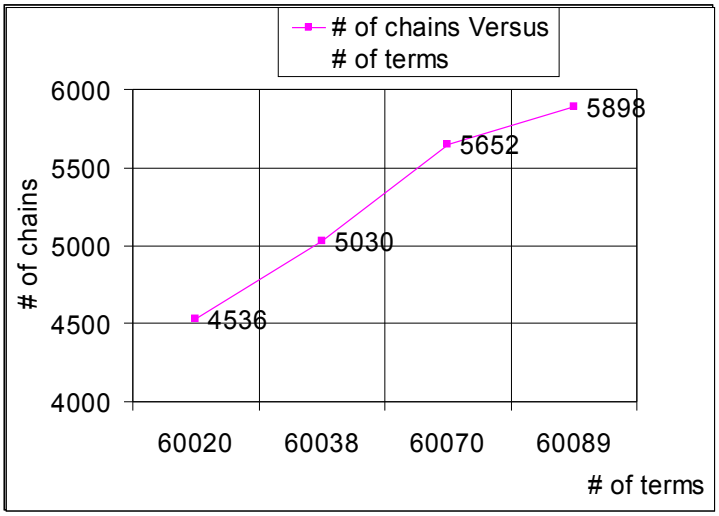
The experiment results are shown in Table 5.1. The results show significant improvements in the quantity of lexical chains, with mild improvements in the quality of the chains. By quantity, we mean the number of lexical chains generated. The size of the ontology between each step increased by an average of 0.05%, while the increase in the number of chains was 10.9%, 12.3%, and 4.3% respectively, very significant compared to the trivial change of the ontology's size. In terms of quality of lexical chains, we looked at two factors: 'strength' and 'richness' of a lexical chain. The strength of a lexical chain is indicated by the length of the chain, and the richness of a lexical chain is indicated by the lemmas of the chain.

The length of a lexical chain represents the degree of importance of a topic (i.e., theme) in the text, in that the author feels the need to stress the topic by repeatedly using closely related terms within a single paragraph to describe the topic. We can reasonably assume that the longer a chain, the more important the theme represented by the overriding sense of the chain. We determined that the average length of a chain decreased by an average of 0.76% between each step.

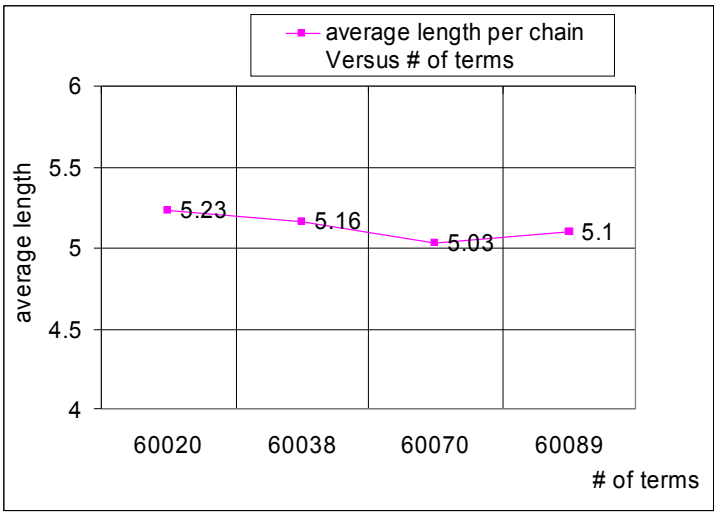
In terms of richness, the lemmas increased by an average of 5.9% between each step, rather significant compared to changes in the ontology's size. We argue that a larger number of unique terms in a chain will provide more valuable information and thus stronger 'evidence' for protein-protein–interaction validity. Figure 5.2 shows the performance metrics versus the number of terms in the ontology.

Measurement	**GO**	**GO + Method**	**GO + Method + IP**	**GO + Method + IP + MF**
# of terms	60020	60038	60070	60089
# of chains	4536	5030	5652	5898
# of chains per doc	45.36	50.30	56.52	58.98
# of chains per para	1.84	2.04	2.30	2.40
average length	5.23	5.16	5.03	5.10
average span	7.44	7.46	7.26	7.46
average lemmas	1.19	1.20	1.29	1.40
average density	0.62	0.61	0.61	0.60

Table 5.1: The experiment results (GO = the Gene Ontology, IP = Interaction Property, MF = Molecular Function)

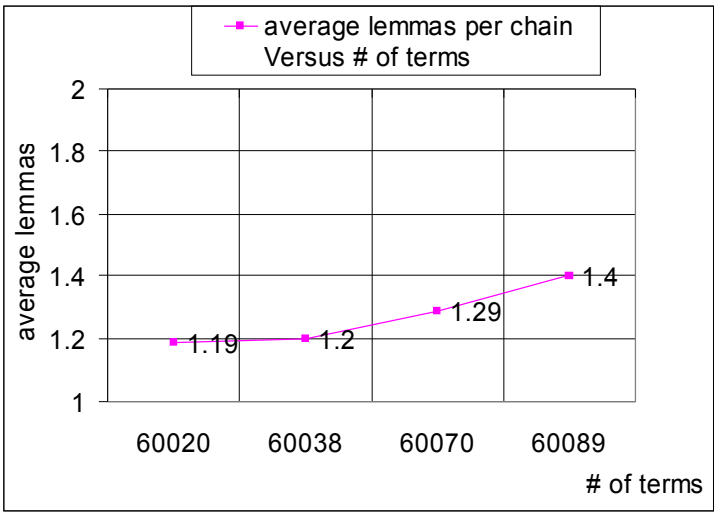


(a) # of chains versus # of terms



(b) Average length versus # of terms

Figure 5.2: Performance metrics of PPIWordNet



(a) Average lemmas versus # of terms

Figure 5.3: Performance metrics of PPIWordNet (continued)

5.3.2 Case study of lexical chains

To investigate the discourse structure of protein-protein–interaction texts, we manually analyzed the lexical chains generated from a randomly selected article [13]. The lexical chains generated are shown in Table 5.2. The chains are grouped by their topic (overriding sense), and the unique terms of the longest chains of the topic are shown in the 'Unique Terms' column. 'Paragraph' column lists all the paragraphs where the chains of the topic are found, while 'Category' shows the category of the topic in PPIWordNet. There were a total of 27 chains found in the article, and 11 unique topics.

The first result that may be observed is the overwhelming number of chains with topics "protein" and "binding". This could be because the focus of the sample text is on protein binding. Secondly, the lexical chains cover a wide range of topics from biological processes to molecular functions. To illustrate the distribution of lexical chains in the text, paragraph 4, which has the largest number of chains extracted, is shown below (different-themed chains are shown in different fonts.) As may be observed in this sample, the topics in the this passage of text were well covered by the lexical chains extracted . However, most of the paragraphs in this article have

Topic	**Paragraphs**	**Unique Terms**	**Category**
protein	2,3,6,7,10,12, 13,14,15,17	{protein}	
binding	2,7,8,9,10,13,14	{binding, binding assay}	molecular function
transport	3	{transport, secretion}	biological process
protein transport	4	{protein transport, protein}	biological process
membrane	4	{plasma membrane, membrane}	cellular component
cytoskeletion	4	{actin cytoskeleton, cytoskeleton}	cellular component
formation	4	{formation}	biological process
transcription	4	{transcription}	biological process
catalytic activity	15	{binding, catalytic activity}	molecular function
growth	18	{growth cone, growth}	biological process

Table 5.2: Lexical-chaining analysis of article "A Conserved Binding Motif Defines Numerous Candidate Target Proteins for Both Cdc42 and Rac GTPases [13]"

only one or two lexical chains generated, in line with the average number of chains per paragraph. It is obvious that more chains or information need to be extracted to enable the system to capture accurately the topics of text.

> Members of the Ras superfamily of small GTPases play a wide variety of cellular signaling roles that mediate proliferation and differentiation, **cytoskeletal** organization, *protein transport*, and secretion. The Ras GTPases have been studied most thoroughly, and now several components of the Ras signaling pathway have been identified using a combination of biochemical and genetic approaches (1, 2). A related family of GTPases, the Rho subfamily, consists of three Rho genes, two Rac genes, Cdc42 and its close homologue G25K, rhoG, and TC10 (3). Early work in Saccharomyces cerevisiae, identified CDC42Sc as a *protein* required for bud emergence (4, 5). In mammalian cells, the Rho subfamily members control the polymerization of actin and the assembly of focal complexes at the plasma membrane in response to extracellular signals (3, 6). For example, microinjection of Rho into serum-starved Swiss 3T3 cells rapidly stimulates stress fiber and focal adhesion ***formation*** (7), while Rac induces membrane ruffles (8) and Cdc42 induces the ***formation*** of filopodia (9). In addition to their effects on the **actin cytoskeleton**, Rho GTPases also have a role in regulating kinase signaling pathways. For example, Rho, Rac, and Cdc42 stimulate a novel nuclear signaling pathway leading to **transcriptional** activation of the serum response element (10). Rac and Cdc42, but not Rho, have also been shown to activate the c-Jun amino-terminal kinase (JNK) signaling pathway leading to c-Jun **transcriptional** activation (11, 12). The mechanisms by which the Rho subfamily of GTPases regulate these apparently diverse biological processes is still not clear.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we have made several contributions to the study of protein-protein–interaction information extraction. We have presented a method for biomedical information extraction that makes use of the lexical-chaining structure in scientific articles to extract strings of biologically related words in protein-interaction contexts. We have developed a hybrid approach for constructing a domain-specific ontology of significant terms extracted from a domain-specific text corpus using statistical information retrieval methods. We have shown through an experiment that the domain-specific ontology has a positive impact on the lexical chains created, and that the extracted strings of semantically related words provide valuable additional information regarding protein-protein interactions.

6.2 Future Work

There are several interesting problems arising from our study that we plan to investigate in future. The immediate direction points to the fine-tuning of the lexical chaining algorithm. Other interesting directions include improving PPIWordNet and judging the qualities of protein-protein interactions using the information extracted.

6.2.1 Lexical chaining algorithm

There are obvious problems with our lexical-chaining algorithm. The first is the way that semantic relatedness is currently calculated, as shown in Table 3.2. The semantic relatedness function defines terms to be related only when they are either in the same synset, siblings, or a parent-child pair. This significantly limits the 'richness' of the chains created. We believe Hirst and St.-Onge's idea of semantic relatedness has more applicable value in real-world applications, as they consider terms as semantically related if their synsets in an ontology are connected by a path that "is not too long" and that "does not change direction too often". We intend to explore this idea and develop a semantic relatedness measure function more applicable to our particular problem. The second problem is the need for text pre-processing. The articles in our corpus were originally available only in HTML format. It was difficult and often error-prone to detect paragraph boundaries in the corpora given their current formatting, which could result in incorrect chains being created.

6.2.2 PPIWordNet

A significant amount of work can be done in this direction. As observed in our experiments, the additions of domain-specific vocabulary increased the number of chains created and the average number of unique terms in a chain, but decreased the average length of chains, even though only slightly. Additional experiments on larger-sized corpora would be worthwhile to increase the amount of available data, and to thus increase the significance of the experiments. As the current PPIWordNet is only a pilot project for our research, we will proceed to refine our methodology and expand PPIWordNet iteratively.

6.2.3 Judging the quality of protein-protein interactions

In order to evaluate protein-protein interactions, more study on metrics for evaluating lexical chains is needed. To begin with, the current methods for computing metrics 'strength' and 'richness' are too simplified. Without a good definition for deep understanding of these metrics, it is difficult to use them to evaluate the quality of protein-protein interactions. Also more metrics need to be invented, i.e., the 'discriminating topics (themes)' of chains that differentiate true positive samples from false positive samples; the 'information content' of lexical chains that reflect

what information contained in a chain ('topic' is the simplest form of information content).

Appendix A

Appendix

The following figures are the two subsets of PPIWordNet: Method and Interaction Property.

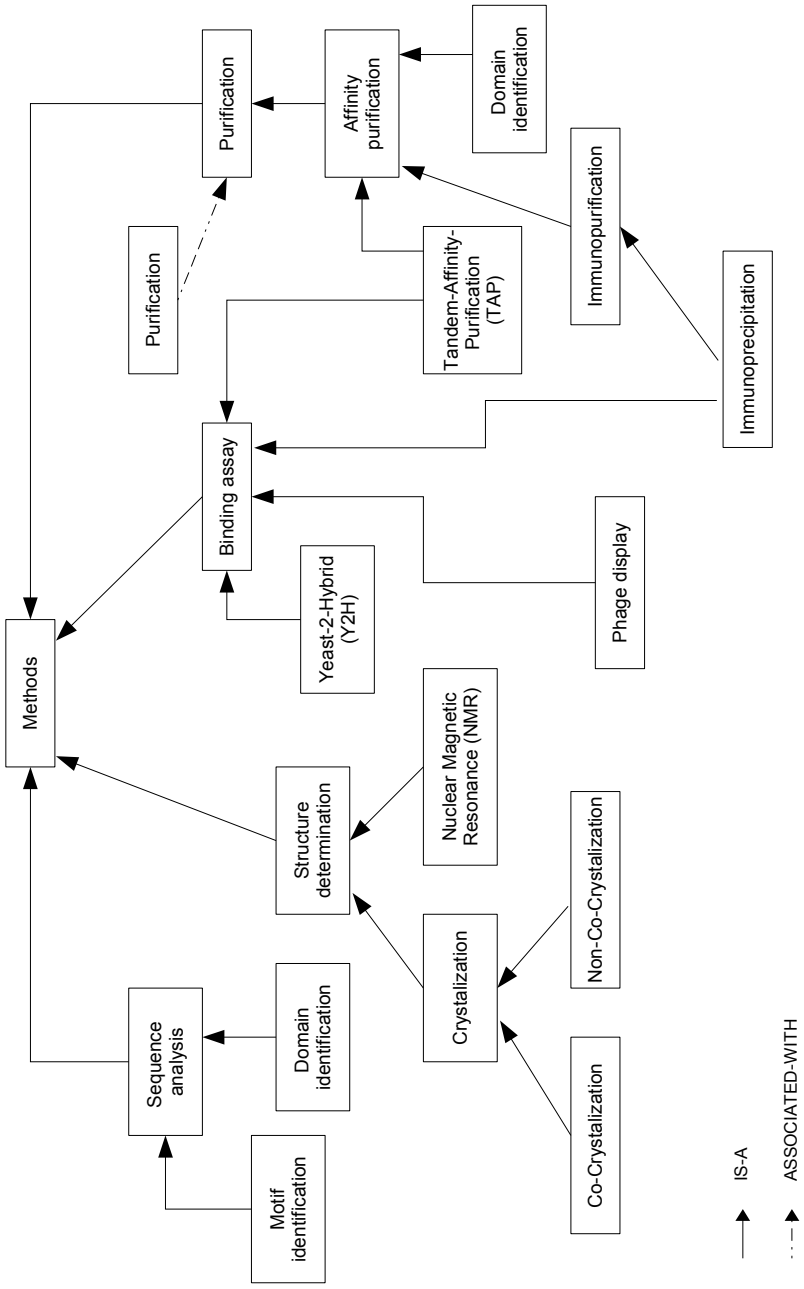


Figure A.1: The final ontology for PPI method terms

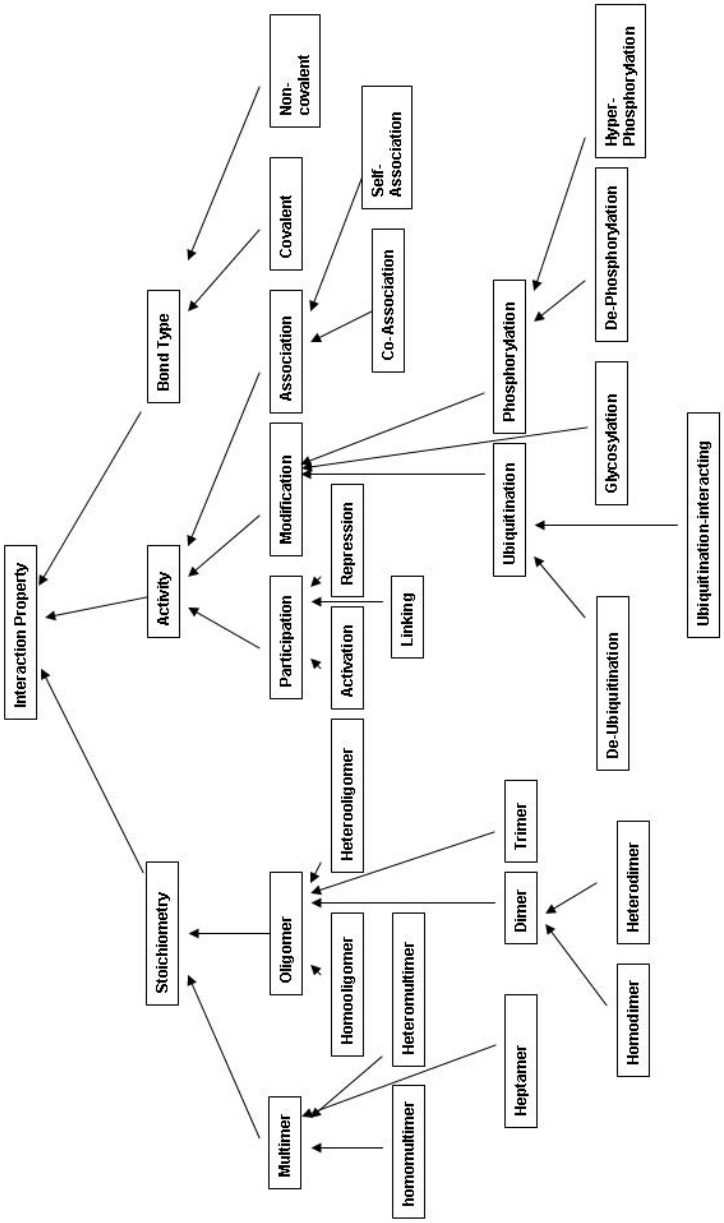


Figure A.2: The final ontology for PPI interaction property terms

Bibliography

[1] BioCreAtIvE. http://biocreative.sourceforge.net/index.html.

[2] UMLS: The Unified Medical Language System. http://umlsinfo.nlm.nih.gov.

[3] WordNet. http://wordnet.princeton.edu/.

[4] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the WWW. *In Proceedings of the First Workshop on Ontology Learning OL-2000, the 14th European Conference on Artificial Intelligence ECAI-2000*, 2000.

[5] S.F. Altschul, W. Gish, W. Miller, E. W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, pages 215:403–410, 1990.

[6] G.D. Bader and C.W.V. Hogue. BIND – a data specification for storing and describing biomolecular interactions, molecular complexes and pathways. *Bioinformatics*, 16(5):465–477, 2000.

[7] Amos Bairoch and Rolf Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45–48, 2000.

[8] S. Banerjee and T. Pedersen. An adapted LESK algorithm for word sense disambiguation using WordNet. *In Proceedings, Fourth International Conference on Computational Linguistics and Intelligent Text Processing*, 2002.

[9] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. *In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS-97)*, 1997.

[10] Dennis A. Benson, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, Barbara A. Rapp, and David L. Wheeler. Genbank. *Nucleic Acids Research*, 28(1):15–18, 2000.

[11] G. Bisson, C. Ndellec, and D. Caamero. Designing clustering methods for ontology building: The MOK workbench. *In Proceedings of the First Workshop on Ontology Learning OL-2000, the 14th European Conference on Artificial Intelligence ECAI-2000*, 2000.

[12] C. Blaschke, M.A. Andrade, C. Ouzounis, and A. Valencia. Automatic extrac-

tion of biological information from scientific text: Protein-protein interactions. volume 7, pages 77–86, 1999.

[13] P.D. Burbelo, D. Drechsel, and A. Hall. A conserved binding motif defines numerous candidate target proteins for both cdc42 and rac gtpases. *The Journal of Biological Chemistry*, 270:2907129074, Dec 1995.

[14] R. L. Chapman. *Roget's International Thesaurus (5th edition)*. Harper Collins, 1992.

[15] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning taxonomic relations from heterogeneous evidence. *In Proceedings of ECAI 2004 Workshop on Ontology Learning and Population.*, 2004.

[16] A.M. Cohen. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. *In Proceedings of the BioLINK 2005 Workshop*, 2005.

[17] A.M. Cohen and W.R. Hersh. A survey of current work in biomedical text mining. *Briefing in Bioinformatics*, 6(1):57–71, Mar 2005.

[18] Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32:D258–D261, 2004. http://www.geneontology.org.

[19] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press New York, NY, USA, 1999.

[20] Matthew John Reinhard Enss. An investigation of word sense disambiguation for improving lexical chaining. Master's thesis, University of Waterloo, Waterloo, Ontario, 2006.

[21] Christiane Fellbaum. *Introduction, WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, Massachusetts, 1998.

[22] 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:S74–S82, 2001(Suppl 1).

[23] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. *In Proceedings of the 3rd Pacific Symposium on Biocomputing*, pages 707–718, 1998.

[24] R. Gaizauskas, G. Demetrious, P.J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The PASTA system. *Bioinformatics*, 19(1):135–143, Jan 2003.

[25] S.J. Green. *Automatically generating hypertext by computing semantic similarity*. PhD thesis, University of Toronto, 1997.

[26] M.A.K. Halliday and R. Hasan. *Cohesion in English*. Longman Group Ltd., 1976.

[27] M. Hearst. What is text mining. *http://www.sims.berkeley.edu/ hearst/textmining*.

[28] Graeme Hirst and David St.-Onge. *Lexical chains as representation of context for the detection and correction of malapropisms*. In Fellbaum, The MIT Press, Cambridge, Massachusetts, 1998.

[29] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *In Proceedings, International Conference on Research in Computational Linguistics*, 1997.

[30] Daniel Jurafsky and James H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 1st edition*. Prentice Hall PTR, 2000.

[31] Z. Kou, W. Cohen, and R. Murphy. High-recall protein entity recognition using a dictionary. *Bioinformatics*, 21(Suppl 1):i266–i273, Jun 2005.

[32] Michael Krauthammer, Andrey Rzhetsky, Pavel Morozov, and Carol Friedman. Using BLAST for identifying gene and protein names in journal articles. *Gene*, 259(1-2):245–252, December 2000. http://dx.doi.org/10.1016/S0378-1119(00)00431-5.

[33] A. Maedche and S. Staab. Discovering conceptual relations from text. *In Proceedings of ECAI2000*, pages 321–325, 2001.

[34] E.M. Marcotte, I. Xenarios, and D. Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17(4):359–363, Apr 2001.

[35] S. Mika and B. Rost. Protein names precisely peeled off free text. *Bioinformatics*, 20(Suppl 1):i241–i247, Aug 2004.

[36] Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–43, 1991.

[37] M. Narayanaswamy and K.E. Ravikumar. A biological named entity recognizer. *In Proceedings of the 8th Pacific Symposium on Biocomputing*, pages 427–438, 2003.

[38] Robert Neches, Richard Fikes, Tim Finin, Tom Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36–56, 1991.

[39] J. Pustejovsky and J.M. Castaño. Robust relational parsing over biomedical literature: Extracting inhibit relations. *In Proceedings of the Pacific Symposium on Biocomputing*, pages 362–373, 2002.

[40] J. Pustejovsky, J.M. Castaño, B. Cochran, M. Kotecki, M. Morrell, and A. Rumshisky. Linguistic knowledge extraction from MEDLINE: Automatic construction of an acronym database. *10th World Congress on Health and Medical Informatics (MEDINFO)*, 2001.

[41] Stuart J. Russel and Peter Norvig. *Artificial intelligence: A modern approach*.

Englewood Cliffs, NJ: Prentice Hall, 1995.

[42] A. Rzhetsky, T. Koike, S. Kalachikov, S. Gomez, M. Krauthammer, S. Kaplan, P. Kra, J. Russo, and C. Friedman. A knowledge model for analysis and simulation of regulatory networks. *Bioinformatics*, 16(12):1120–1128, Dec 2000.

[43] R. Saito, H. Suzuki, and Y. Hayashizaki. Construction of reliable protein-protein interaction network with a new interaction generality measure. *Bioinformatics*, 19(6):756–763, Apr 2003.

[44] T. Sekimizu, S. Park Hyun, and Jun'ichi Tsujii. Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. *Genome Informatics*, 9:62–71, 1998.

[45] H.G. Silber and K.F. McCoy. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28:487–496, 2002.

[46] M. Stairmand. *A computational analysis of lexical cohesion with applications in information retrieval*. PhD thesis, University of Manchester Institute of Science and Technology, 1996.

[47] B.J. Stapley and G. Benoit. Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in MEDLINE abstracts. *In Proceedings of the 5th Pacific Symposium on Biocomputing*, pages 529–540, 2000.

[48] 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*, pages 541–553, 2000.

[49] Mike Uschold and Martin King. Towards a methodology for building ontologies. *In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95*, 1995.

[50] I. Xenarios, D.W. Rice, L. Salwinski, M.K. Baron, E.M. Marcotte, and D. Eisenberg. DIP: the database of interacting proteins. *Nucleic Acids Research*, 28(1):289–291, 2000.

[51] H. Yu and E. Agichtein. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19(Suppl 1):i340–349, 2003.

[52] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 21(7):1178–1190, May 2004.