CONTEXT SPECIFIC TEXT MINING FOR ANNOTATING PROTEIN INTERACTIONS WITH EXPERIMENTAL EVIDENCE

YOGESH PANDIT

Submitted to the faculty of the School of Informatics

in partial fulfillment of the requirements

for the degree of

Master of Science in Bioinformatics,

Indiana University

May 2013

Accepted by the Faculty of Indiana University,

in partial fulfillment of the requirements for the degree of

Master of Science in Bioinformatics

Master's Thesis

Committee

_____

Mathew Palakal, PhD, Chair

_____

Yunlong Liu, PhD

_____

Xiaowen Liu, PhD

Dedicated to my Parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would never have been able to finish my dissertation without the guidance of my advisor, colleagues, help from friends, and support from my family.

I would like to express my deepest gratitude to my advisor, Dr. Mathew Palakal, for his guidance. He has unstintingly shared his excitement about and knowledge of biomedical literature mining. I am extremely grateful to him for his patience with me throughout the time course of this project. I am thankful to him for the opportunity and the freedom he provided to pursue something interesting.

I would also like to thank Dr. Yunlong Liu and Dr. Xiaowen Liu for graciously accepting to be on my committee.

I want to thank my fellow lab mates. Especially, Deepali Jhamb for help at every step and Tulip Nandu who helped me with the initial annotation of the data. I am grateful to Dr. Meeta Pradhan for her support and constant encouragement.

Work would not have been so much fun without my friends and roommates. I am extremely grateful to them all.

Finally, I would like to thank my family, and my friend Akshita Dutta. They have always been there cheering me up and stood by me through the good times and bad.

ABSTRACT

YOGESH PRAKASH PANDIT

CONTEXT SPECIFIC TEXT MINING FOR ANNOTATING PROTEIN INTERACTIONS WITH EXPERIMENTAL METHODS

Proteins are the building blocks in a biological system. They interact with other proteins to make unique biological phenomenon. Protein-protein interactions play a valuable role in understanding the molecular mechanisms occurring in any biological system. Protein interaction databases are a rich source on protein interaction related information. They gather large amounts of information from published literature to enrich their data. Expert curators put in most of these efforts manually. The amount of accessible and publicly available literature is growing very rapidly. Manual annotation is a time consuming process. And with the rate at which available information is growing, it cannot be dealt with only manual curation. There need to be tools to process this huge amounts of data to bring out valuable gist than can help curators proceed faster. In case of extracting protein-protein interaction evidences from literature, just a mere mention of a certain protein by look-up approaches cannot help validate the interaction. Supporting protein interaction information with experimental evidence can help this cause. In this study, we are applying machine learning based classification techniques to classify and given protein interaction related document into an interaction detection method. We use biological attributes and experimental factors, different combination of which define any particular

interaction detection method. Then using predicted detection methods, proteins identified using named entity recognition techniques and decomposing the parts-of-speech composition we search for sentences with experimental evidence for a protein-protein interaction. We report an accuracy of 75.1% with a F-score of 47.6% on a dataset containing 2035 training documents and 300 test documents.

CHAPTER ONE: INTRODUCTION

Literature is a very popular mode of publication of research findings, information, news, data etc. With the ease of publishing on the Internet, the numbers are ever increasing. There are journals covering all kinds of scientific/ research oriented fields. PubMed alone has grown to hold over 22 million citations. The free full-text branch out of PubMed, PubMed Central archives over 2.6 million articles. All this adds up to the wealth of publicly available information. Such vast amount of data makes it a challenge to develop norms or standards of individual data elements. Even today, biological databases greatly rely on expert curators for manual extraction of valuable information. To automate process of manual curation to some extent, to identify data points and coagulate documents falling under similar context, it is very important to develop intelligent and efficient text mining systems. [1]

Biomedical literature can be harvested to extracted information pertaining to diverse contexts. The most elementary form of biological attributes that can be extracted are biological entities like genes, proteins, chemicals, organisms, strains and more. The more complex techniques involve compressing full-text articles to a set of few representative sentences, inferring gene and protein type of ontology, inferring actions of drugs under certain biological conditions etc. Another such technique is extraction of protein interactions from literature. Proteins are the building block and they interact with other proteins to make a unique biological phenomenon. Protein-protein interactions are valuable to understand and interpret the molecular mechanisms governing a biological

system. An example would be the interaction of *BRCA1* with *BARD1*. However, a mutation of *BRCA1* can disrupt the interaction, which can lead to breast cancer [2].

Motivation

There are many protein interaction databases available in the research community. Each database specializes in annotating and maintaining data on certain type of interactions. BioGRID [3] hold around 630,000 interactions from model organisms and humans. These interactions are curated from high-throughput datasets that are derived from 37,000 publications. A team of 14 curators manually curates this vast amount of literature. Another such database is MINT (Molecular interaction database) [4]. MINT has 241,458 interactions on 35509 proteins curated from 5516 scientific articles by a team of 7 researchers. IntAct [5] is a protein interaction database maintained by the European Bioinformatics Institute. As of September 2011, it contains 275,000 interactions curated from 5000 publications. IntAct also accepts direct user submissions to add to their protein interactions database. The Human Protein Reference Database (HPRD) [6] is a widely used database pertaining to human protein. It contains information on protein entries, protein-protein interactions, post-translational modifications, protein expression, subcellular localizations and domains. The number of protein-protein interactions it holds, as of April 2013, is 41,327. HPRD states in their FAQs that they do not make use of any literature mining algorithms and rely on expert biologists for manual curation. There are many more such database that make use of the scientific literature for identification of protein interaction evidences. As evident from the statistics, there is a gap in the demand to supply ratio. The amount of available literature is rapidly

increasing. A simple search on PubMed with the terms "protein protein interactions" returns about 275,000 entries. So far number of publications manually curated by all the databases put together is not more than 60,000. The not annotated publications contain valuable information that can be useful to the scientific community. This is our main motivation behind pursuing this research topic. We do not intend to replace the expert curators. However, we belief, with the efficient use of text mining technology we can help speed up the task of manual curation. Text mining techniques can identify evidences of protein-protein interactions along with the interaction detection method used to identify that particular interacting pair. This can significantly cut down the laborious task of manually going though each and every paper to identify the context and the interacting protein pairs. The curators and briefly refer the publication using the information extracted about the interacting protein pairs and the corresponding interaction detection method. A summary of number of interactions identified by manual curation of scientific literature is mentioned in Table 1.

Database	# Interactions	# Articles	# Curators/Team Members
BioGRID	630,000	37,000	14
MINT	241,458	5516	7
IntAct	275,000	5000	-

Table 1: Summary of interactions, articles and number of curators per protein interaction database

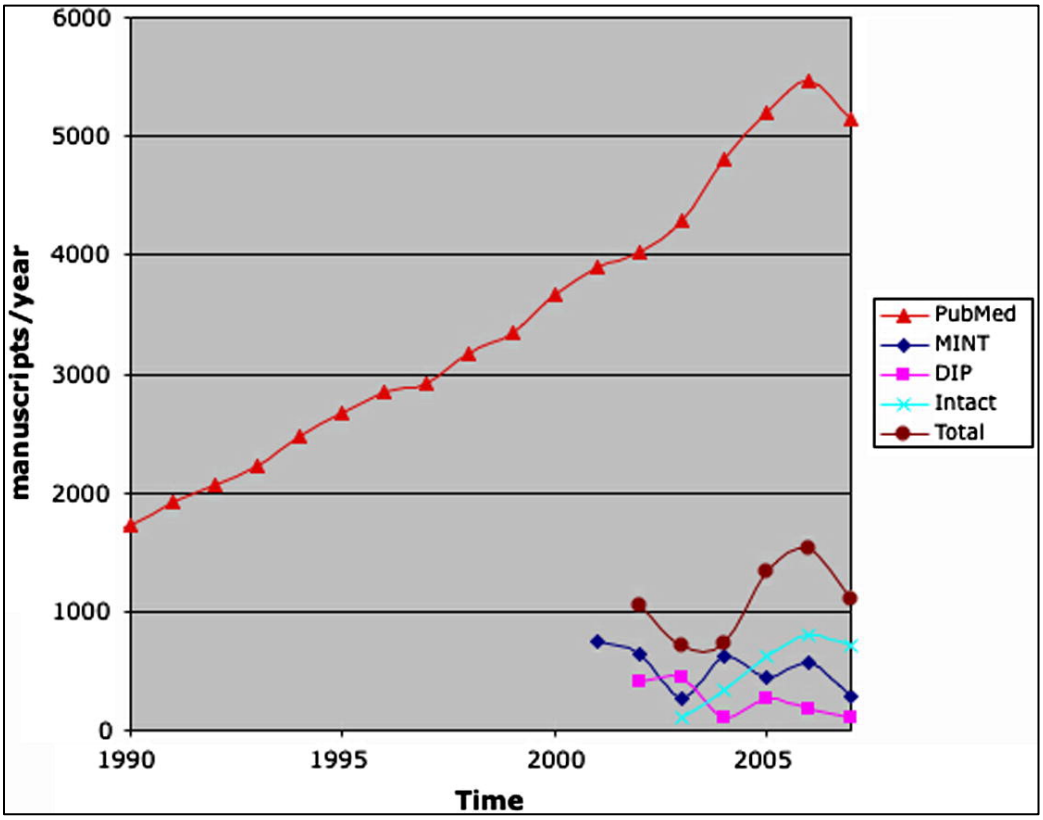

Figure 1: Increased gap between published information and curation by IMEX databases

Figure 1 shows statistics where the estimated number of publications reporting protein interactions each year was estimated by searching PubMed with the keywords "protein interaction". The fraction of retrieved articles containing protein interaction information was approximated by manual scan of 100 abstracts [7]. Figure 1 can help us visualize the growing gap between the published literature and manual curation by protein-protein interaction databases. This gap is significant considering the computational technology available now days. This gap in curation prevents valuable data from hitting the searchable public databases.

Text mining can be referred to as data mining for literature. Literature can be in any form like published scientific literature, web pages, new articles, OCR scanned documents, patents, blogs, textual information on websites pertaining to the topic of interest etc. Typically, text-mining tasks include named entity recognition, clustering, classification/categorization, and sentiment analysis and document summarization. Broadly, these tasks can be categorized as information retrieval, information extraction, pattern recognition and lexical analysis. Natural language processing is used to convert text to data that can be analyzed [8]. Text mining has been of immense importance in the fields of business intelligence, finance forecasts, national security, scientific discovery, sentiment analysis, advertising, question answering, social media monitoring and many more. As an example, IBM recently made public their "deep QA" based product called Watson. It beat human competitors at the game of Jeopardy. Watson makes use of very complicated natural language processing algorithms to understand and answer questions related to any topic on earth. Text mining has also been heavily used to analyze sentiments of voters during election campaigns. Along with such wide range of application, text mining can be applied on biomedical data as well. For example, researchers have utilized text-mining techniques to classify suicide notes into categories of emotions to understand the suicidal patient's thoughts [9–11]. A really interesting application has been a literature search tool where the query can be a chemical structure. The chemical names in the literature databases are converted to a representation called SMILES or InChi and the chemical structure similarity is calculated against the query using the Tanimoto coefficient. Literature documents where some novel compounds that can have structural similarity with existing chemical compounds can be quite easily

pulled up with great precision [12]. Natural language processing is being widely used in clinical decision support. The goal of Clinical Decision Support (CDS) is to "help health professionals make clinical decisions, deal with medical data about patients or with the knowledge of medicine necessary to interpret such data" [13]. Clinical Decision Support Systems are defined as "any software designed to directly aid in clinical decision making in which characteristics of individual patients are matched to a computerized knowledge base for the purpose of generating patient-specific assessments or recommendations that are then presented to clinicians for consideration" [14]. NLP has played a significant role in utilizing free-text information to drive CDS, representing clinical knowledge and CDS interventions in standardized formats, and leveraging clinical narrative [15].

A technique very popularly used in text mining is text categorization or text classification. Few classical examples of text classification are spam filtering, sentiment analysis, language identification etc. In the biomedical context, the sequence labeling can be projected as a classification problem, emotion identification in suicide notes, binary text classification like cancer related or not. The problems like spam/no-spam or cancer/not-cancer or positive/negative sentiment are binary classifications, which means data can be classified into either of the two classes. The features set is consistent throughout the dataset. A Naïve Bayes classifier has proven to perform well in such cases, where given certain features in the email it calculates the conditional probability of it being spam or not [16], [17]. Spam classifiers have evolved to be self-learning and adaptive in nature where the model is in an incremental training mode [18]. SVM (support vector machine) are supervised learning models that are very widely used

classifiers for data analysis and pattern recognition. Basic SVM learning model performs as a non-probabilistic binary linear classifier. In addition, SVM can be efficiently used for non-linear classification using a trick, called kernel trick where inputs are implicitly mapped to high-dimensional feature spaces. The key is to determine the optimal boundaries between different hyper plane representations of input data space. However, the kernels have issues with capacity control, as all learning is done in terms of dot products between items.

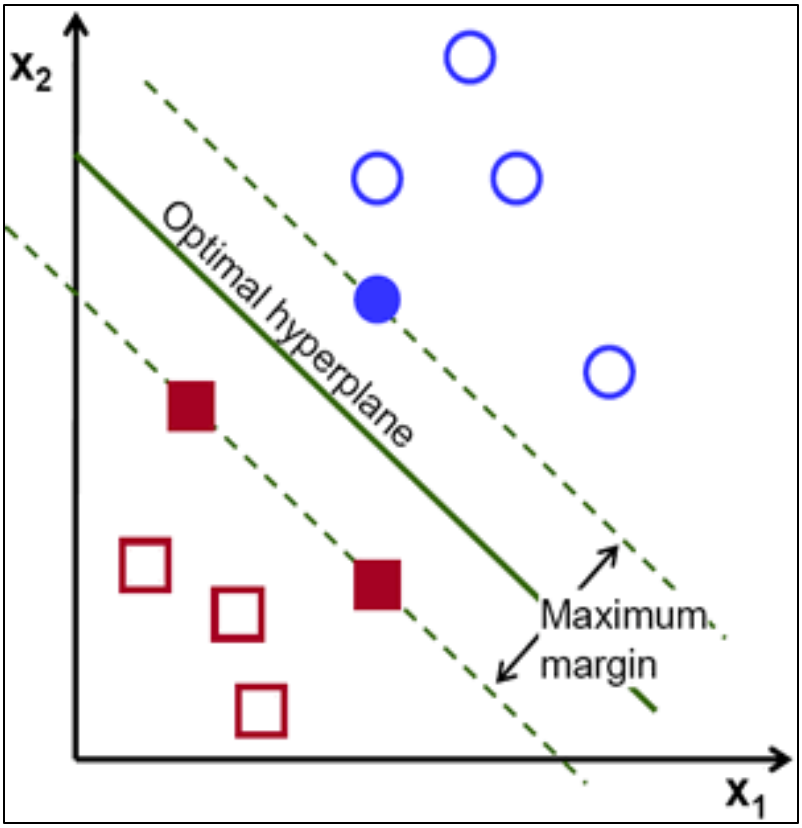

Figure 2: Optimal hyper plane

Figure 2 shows optimal class boundaries for a classifier. A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points.

Then, the operation of the SVM algorithm is based on finding the hyper plane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM's theory. Therefore, the optimal separating hyper plane *maximizes* the margin of the training data [19].

Another classifier is Decision Trees. Decision trees are designed with the use of hierarchical division of input data vectors with the use of different text features. The hierarchical division is designed in order to create class partitions. For a given text instance we determine the partition that it is most likely to belong to, and use it for classification. Neural networks are popular class of classifiers, which are used in a wide variety of domains. A neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Neural networks, SVM are discriminative classifiers unlike Bayesian classifiers, which are generative. The idea behind Bayesian classifiers is to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the documents [20]. All these algorithms are well tested and heavily used. However we decided to use Logistic Regression, which is a discriminative probabilistic classification model for our problem.

In this study, we break down full-text documents into feature vectors, which are formed of experimental factors, key biological entities and some manually identified entities. These features are trained using logistic regression classifier with interaction detection method as class labels. These interaction detection methods are the one's used to identify

protein-protein interactions mentioned in that particular paper. Once we have a trained model, it can take any document as input and predict the interaction detection method that the document might belong to. This is a important and challenging problem to tackle. Efficient implementations can help speed up the manual process of protein-protein interaction curation and annotation from scientific literature.

Context-specific mining

Text mining has its strengths and weaknesses, the most common weakness being the noisy and unspecific data generated as a result of natural language processing (NLP). Hence, context-specific information retrieval is required to circumvent the flaws in traditional text mining approaches. Text mining has proven to be of immense importance in domain specific studies. In drug discovery context, text mining has been applied to extract drug-drug interactions [21][22] and also to explore the network of chemical relations and also in the context of associated binding proteins [23]. Algorithms have been developed to convert chemical names to molecular formula and structures [24][25][26][27]. Such sophisticated use of NLP techniques has helped in building databases for chemical information [28]. Content of the documents has been used for classification into semantic topics [29][30]. NLP techniques have also been applied to interpret emotions behind suicide notes [10].

Protein-protein interactions are manually curated to enrich database like BioGRID [3], MINT [4], DIP [31] and many more. Text mining techniques can however be limiting, as

they cannot efficiently annotate the interaction with the experimental method. The validity and reliability of an interaction can be determined based on the experimental evidences that support it. Augmentation of protein interaction information with the experimental data can enrich the annotation for better analyses. This makes the process of information extraction specific to the context of experimental methods used for protein interactions. In the context of protein-protein interactions mentioned in a particular document, not always all the information will be valuable. Certain sections of a document can be a goldmine for studying, understanding and annotating protein-protein interactions. On the other hand, some sections may be completely irrelevant to the context of protein interactions. With the help of experts in biology we identified that the methods section of a document has most information that can help annotate the interacting protein pair with valuable experimental information. The methods section contains details on the cellines, techniques, media, antibodies, chemicals, temperatures and other such data that define the working of interaction detection method. We broke this problem into a text classification problem where the input vector is the various different biological attributes identified mainly from the methods section and the class labels are the interaction detection methods. We used the publicly available data on interaction method classification from BioCreative

Experimental evidence

Protein-protein interactions occur when two or more protein bind together in a peptide bond to carry out a biological function. This forms the foundation for the proper functioning of any biological processes. For example, the process of signal transduction is heavily dependent on the interaction between proteins on the exterior of a cell with that

of proteins inside the cell. Modification of proteins can change the protein interaction itself. And it is known that proteins respond to the environment in many ways. protein–protein interactions are of central importance for virtually every process in a living cell. Information about these interactions improves our understanding of diseases and can provide the basis for new therapeutic approaches. Protein–protein interactions are at the core of the entire interactomics system of any living cell [32]. Due to such high importance of protein interactions, there are a multitude of methods used to detect them. Each method has its pros and cons, mainly in regards to the sensitivity and specificity. Like in any evaluation system, high sensitivity means that most of the interactions from real world are detected using the screening techniques. And high specificity indicates that most of the interactions detected by the screening should occur in reality. According to the PSI MI ontology [33] for the interaction detection methods, there are around 115 screening techniques to detect protein interactions. The popularity of a certain method depending on the number of articles published on it is shown in Figure 11. Each method has its own approach at detecting interactions. For example, yeast two-hybrid is a high-throughput screening method that allows for interactions between proteins that are never expressed in the same time and place. This is at the cost of its specificity. Affinity capture mass spectrometry, on the other hand, does not perform in this manner. Yeast two-hybrid data better indicates non-specific tendencies towards sticky interactions rather while affinity capture mass spectrometry better indicates functional in vivo protein–protein interactions [34]. Each method has its own significance and the distinction is vivid from the protocols used and mentioned in literature. These protocols can help identify the

interaction detection methods used to detect the interactions and can be termed as experimental evidences.

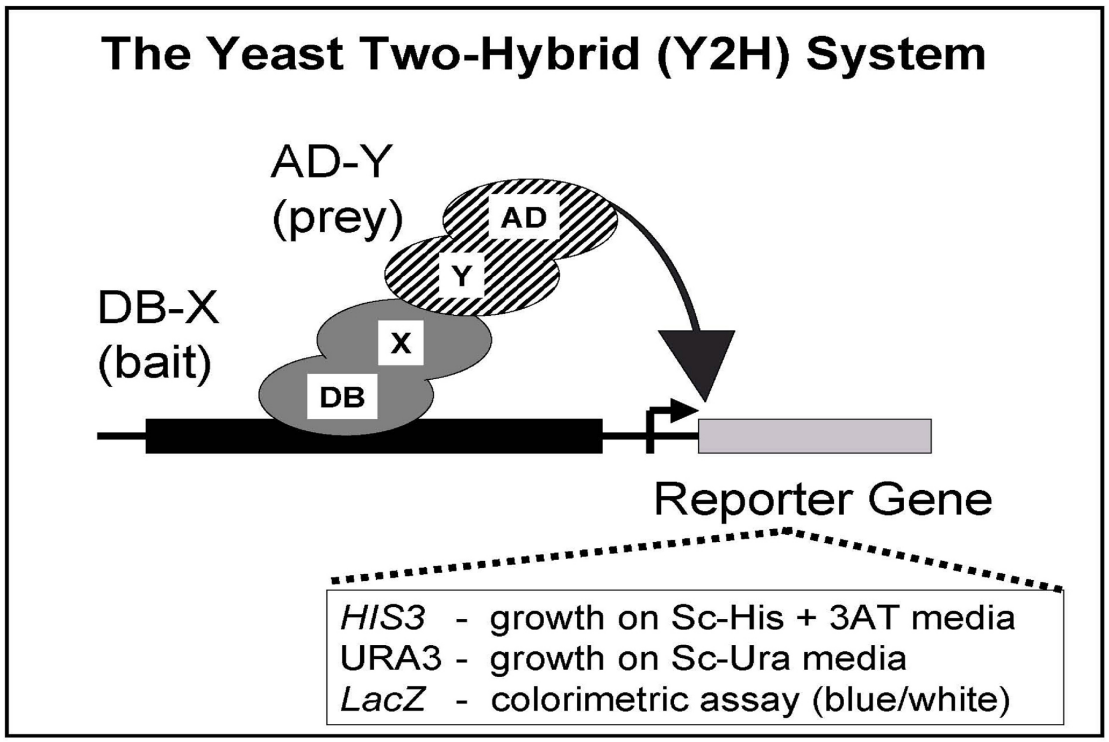

Figure 3: A Yeast Two-Hybrid system

And example of details provided by a detection method in the literature is shown in Figure 3 [35]. The mention of HIS3, URA3, LacZ along with Sc-His + 3AT media, Sc-Ura media indicates that the document is talking about yeast two-hybrid for interaction detection. Another example of how information about detection methods can be mined from the literature can be shown from Figure 4 [36]. This pull down assay schematic shows some of the important details of the interaction detection method itself like agarose bead, affinity ligand, GST and SDS-PAGE. Like mentioned before, this is valuable

information occurring in literature that can help distinguish pull down assays from other interaction screening techniques.

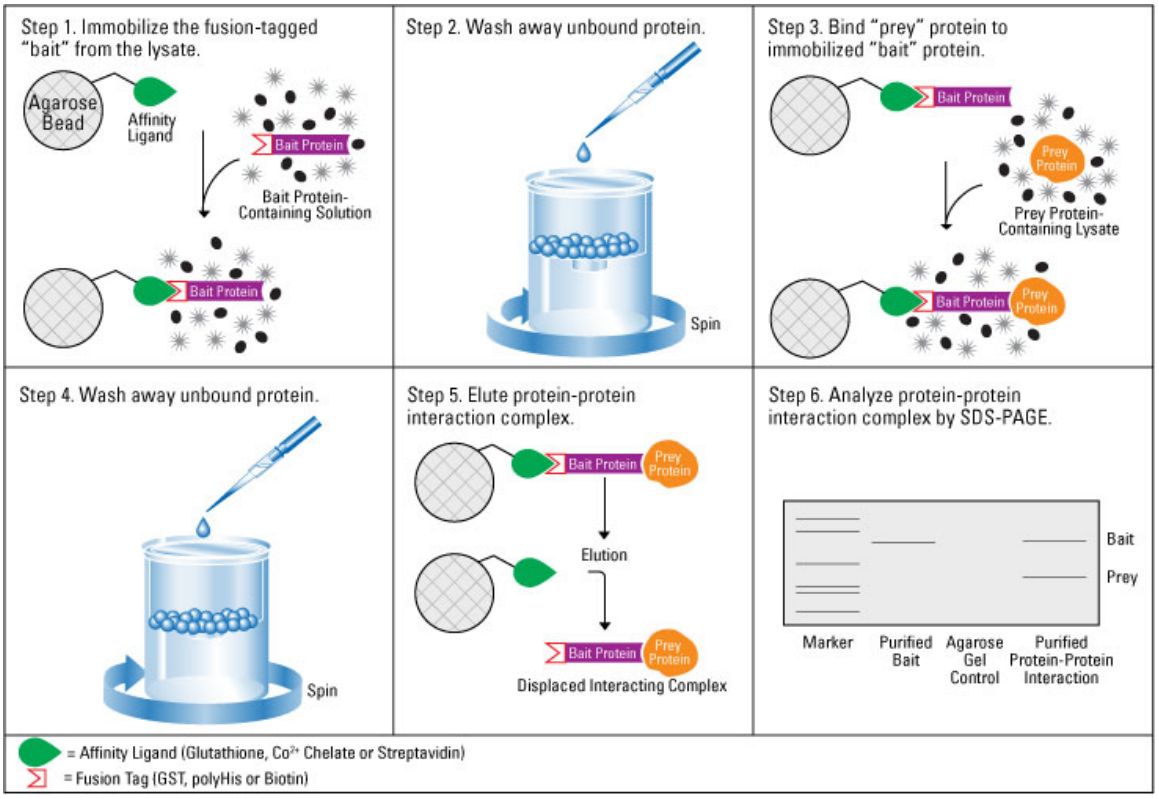

Figure 4: Pull down assay schematic

There are many different tools and databases available for protein-protein interactions few are tabulated in Table 2.

Database	Description	URL
Online Databases storing protein-protein interactions		
BIND	Bimolecular interaction network database contains over 200,000 human curated interactions.	http://bind.ca
DIP	Database of Interacting Proteins enlists experimentally determined 75,400 interactions between proteins covering 571 organisms.	http://dip.doe-mbi.ucla.edu
HPRD	The Human Protein reference database contains interaction networks for each protein in human proteome.	www.hprd.org
HPID	Human Protein Interaction Database combines BIND, DIP and HPRD	www.hpid.org
IntAct	Open source protein interaction database, contains approximately 3,12,217curated binary molecular interactions	http://www.ebi.ac.uk/intact/
MINT	Molecular INTeraction Database stores biological molecule interactions.	http://mint.bio.uniroma2.it/mint
STRING	Consists of known and predicted protein-protein interactions. The database currently covers 5,214,234 proteins from 1133 organisms.	http://string-db.org/
Online protein-protein interaction information extraction systems		
BioRAT	BioRAT is an information extraction tool and search engine for biological research	http://bioinf.cs.ucl.ac.uk/software_downloads/biorat/
GeneWays	GeneWays is an integrated system that combines various sub tasks of extracting, analyzing, visualizing and integrating molecular pathway data	http://anya.igsb.anl.gov/Geneways/GeneWays.html

Table 2: Tools, databases for extraction of protein interactions

Protein interaction curation

Protein interaction databases have been formed with the goal of curating protein and genetic interactions with great details. Such data can help decode the mechanism behind cellular physiology. Availability of such information is ever growing. With the primary goal of reducing curation redundancy and sharing data these databases are federated by International Molecular Exchange (IMEx) consortium. The PSI-MI provides the logic model and the controlled vocabulary for representation of molecular interactions. Not surprisingly, the members of the IMEx consortium themselves are the main contributors to the development and maintenance of the PSI-MI ontology. Sharing data using ontology standardizes the growth of the data.

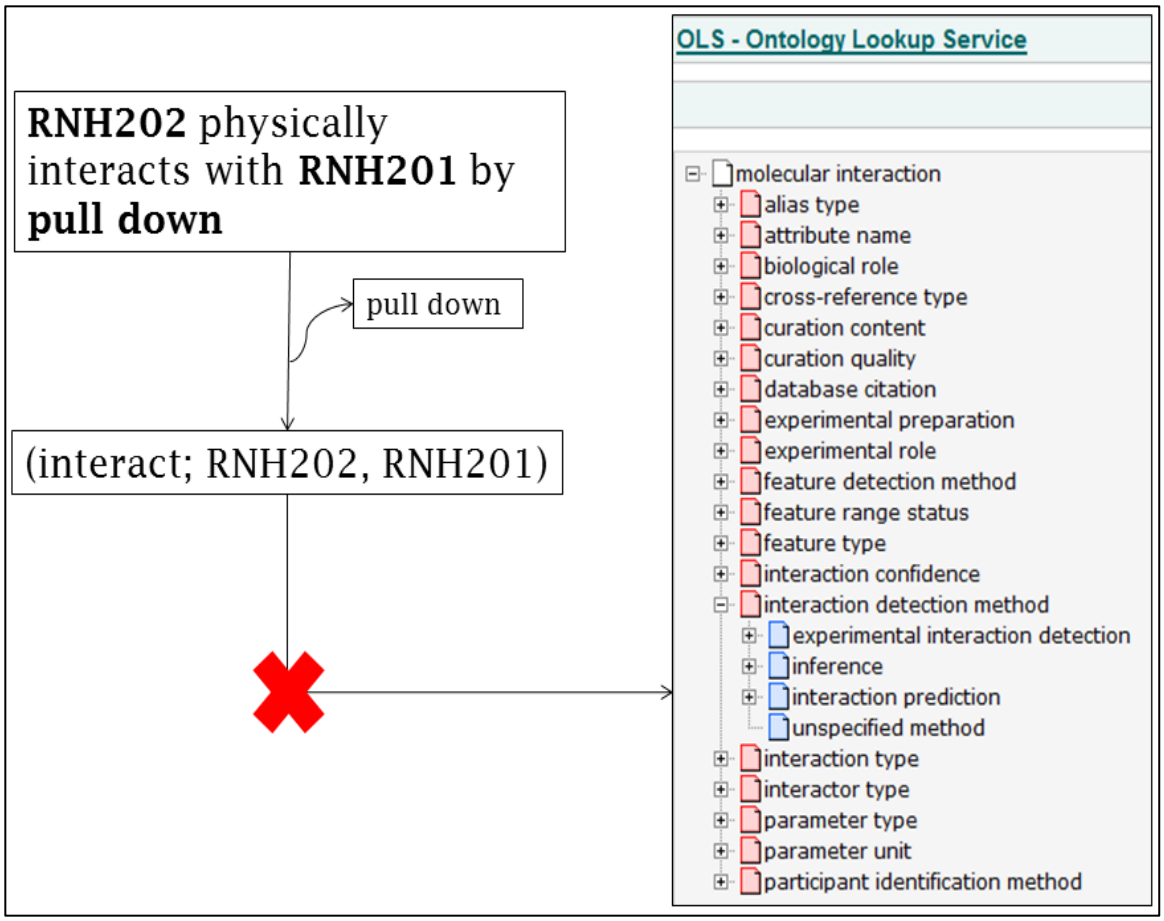

Figure 5: Example to show how protein interactions cannot be linked to ontology for data standardization

Figure 5 is an example that demonstrates how a protein interaction pair extracted from a sentence can be just a binary pair not linkable to the interaction detection method ontology. The protein-protein interaction databases are growing very rapidly. To minimize redundancy, they need to be tagged to the same PSI-MI ontology with a different source. If the experimental evidence is not considered in binary interactions, relating interactions with ontology will not be possible. However, if interacting proteins are annotated with the interaction detection method during automated extraction, it can be linked to the PSI-MI ontology. Figure 6 graphically shows how this can be possible.

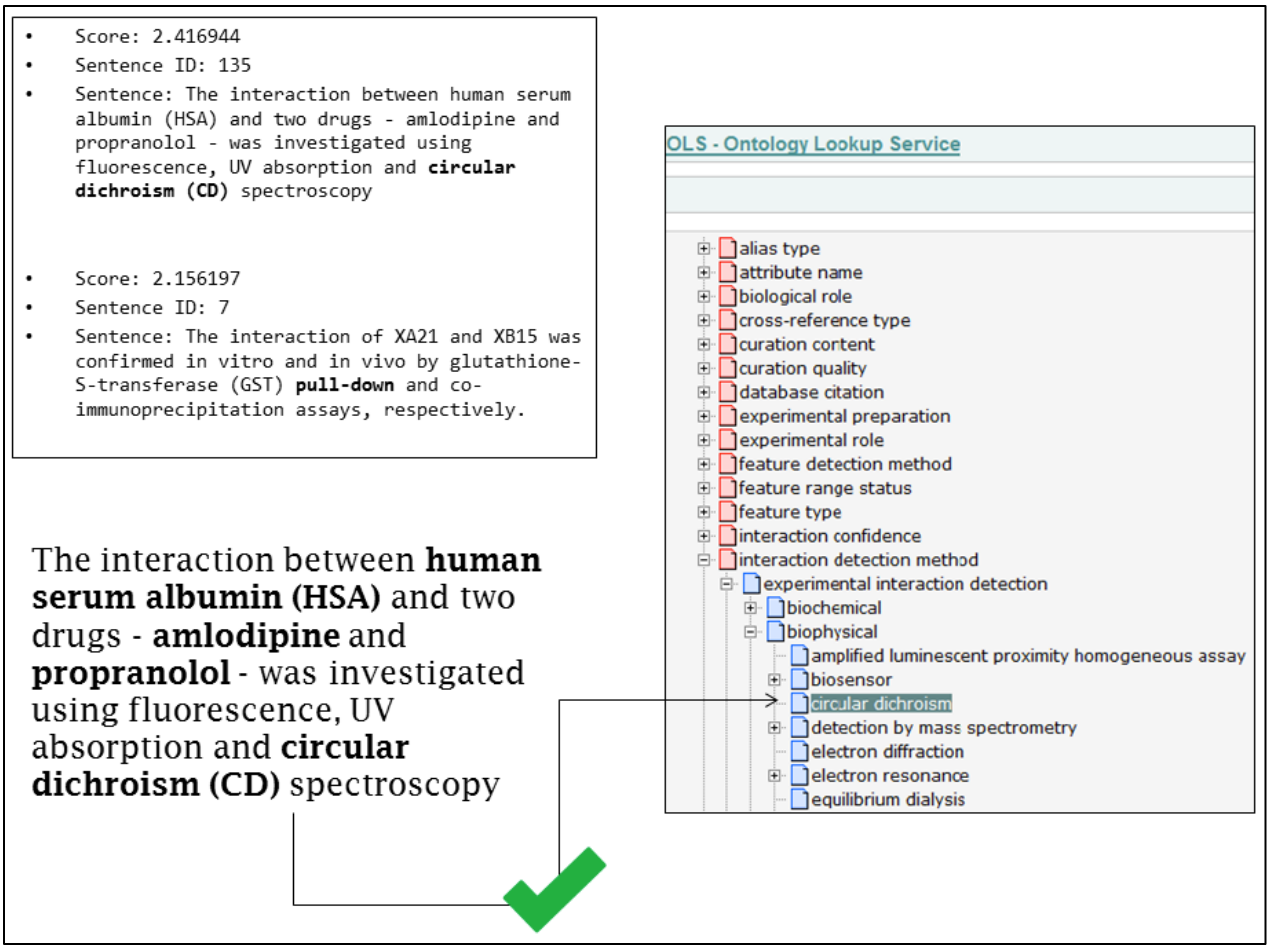

Figure 6: Example showing how protein interactions can be linked to ontology if annotated with interaction detection method

Two important issues with the automatic extraction of information are validation and usability. Any particular biological entity is often discussed in sufficient detail in the article. However, the extraction process usually discards these details and returns only the entities. Protein-protein interactions (PPI) are very commonly extracted information from the published biomedical literature. For example, from a sentence like "As expected from the yeast two-hybrid results, p18Hamlet was able to interact with p38α and also with p38β but not with p38γ and p38δ, or with the p38 activator MKK6" [37], the interaction that is returned is "p18Hamlet interacts with p38α".

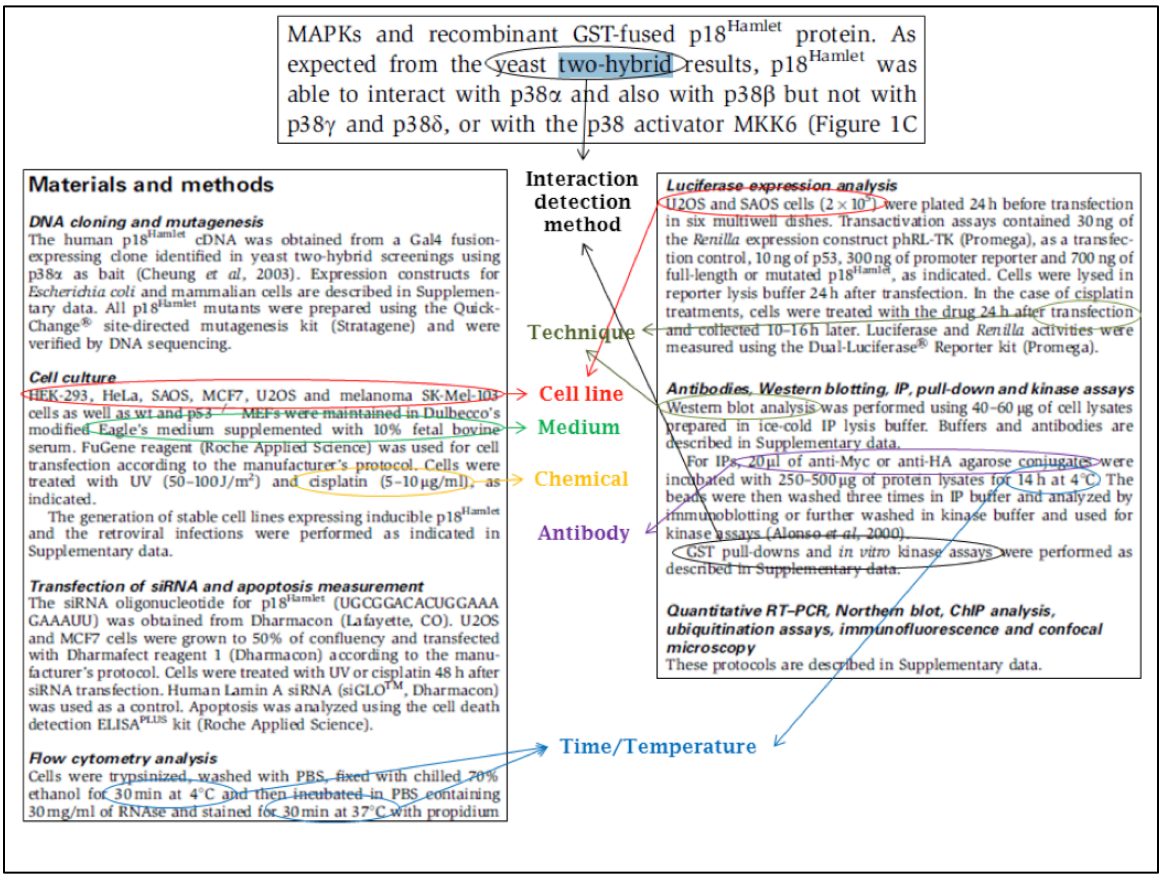

Figure 7: Biological features associated with interaction detection methods

Important information like the protein interaction detection method is discarded. Given a context-specific text mining method there is usually enough information in the articles that can help extract relevant details, based on the biological features. Interaction detection methods are the most popularly used experimental evidences to annotate interactions. PSI-MI [33] ontology is a controlled vocabulary for all the interaction detection methods used in experimentally identifying protein-protein interactions. All protein interaction databases have been annotating according to PSI-MI standards since it

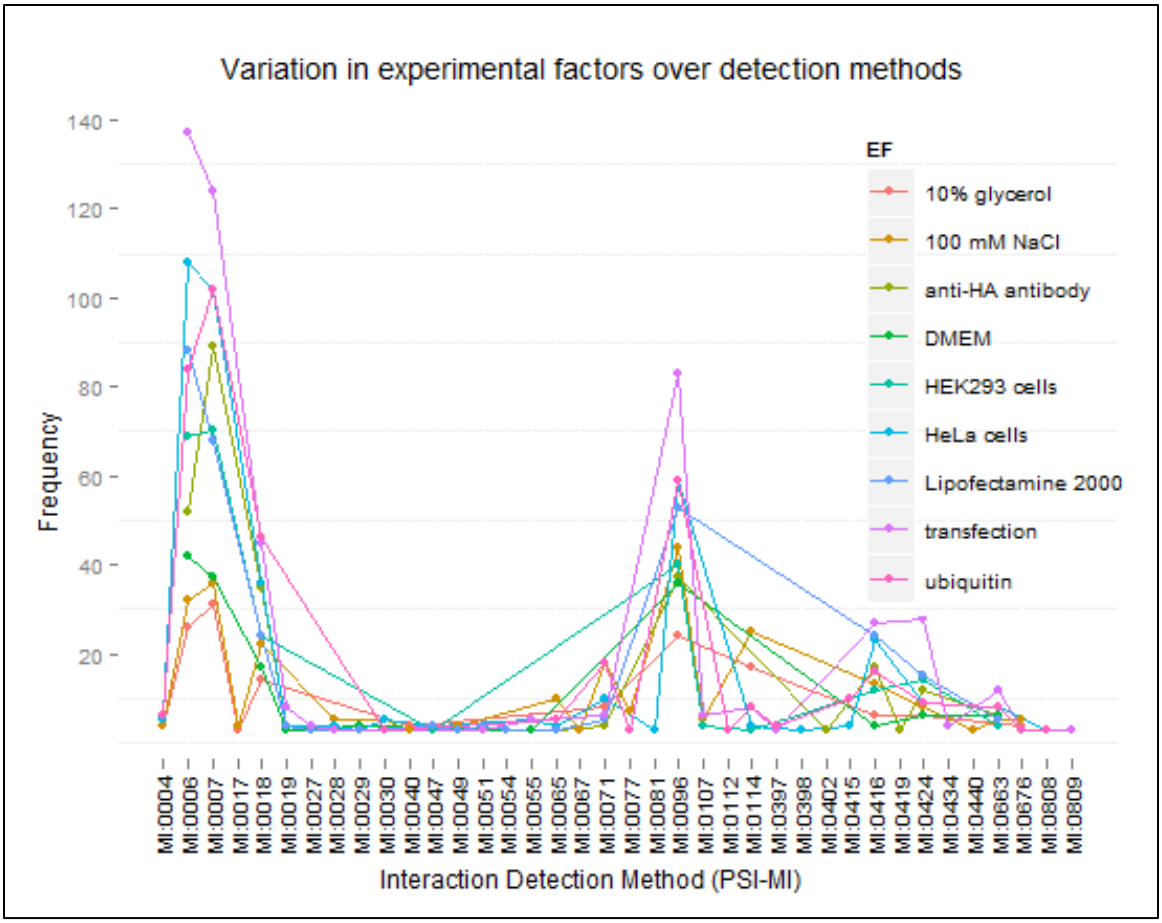

Figure 8: Variation in occurrences of experimental factors per interaction method document

was developed. Figure 7 shows in detail the biological features occurring in the "Methods" section that can be used to associate the interaction detection method with a given protein interaction. Figure 8 highlights how the occurrence of a particular experimental factor varies with documents belonging to different interaction methods. These combinations of varying biological attributes can be used as distinguishing factors between documents falling under certain interaction detection methods.

In this study, we report on the development of an approach to process articles to identify methods that were used to discover the protein-protein interactions. There are many different ways to classify a document. However, if the problem in hand is as specific as annotating protein interactions with experimental methods, a generic document classification approach should not suffice. Not all possible set of features can be specific to the context of the problem. That is why; we make use of features most relevant to the domain of the problem. These features are the experimental factors and biological entities that should be directly related to the standard operating procedures of the interaction detection methods.

CHAPTER TWO: LITERATURE REVIEW

Text mining techniques have evolved and can now return more than just mere co-occurrences of proteins as interactions. There have been many different approaches for rightly extracting binary interactions. Scientific literature can provide insights into novel discoveries and hypotheses from research.

Discovery and extraction of information from free text, encompasses scientific data mining. There are four different categories, for text mining: information retrieval, information extraction, building a knowledgebase and knowledge discovery. In information retrieval, user submits a query to the search engine and receives relevant documents or text, which are fetched based on matching keywords contained in the query, or other scientific metadata (author, title, name of journal and so on) attributes. Information extraction identifies existence of genes or diseases, as well as complex relationship between these entities like protein-protein interactions and gene-disease associations. Knowledge discovery deduces hidden or undiscovered knowledge by applying text-mining algorithms to the data extracted from literature [38].

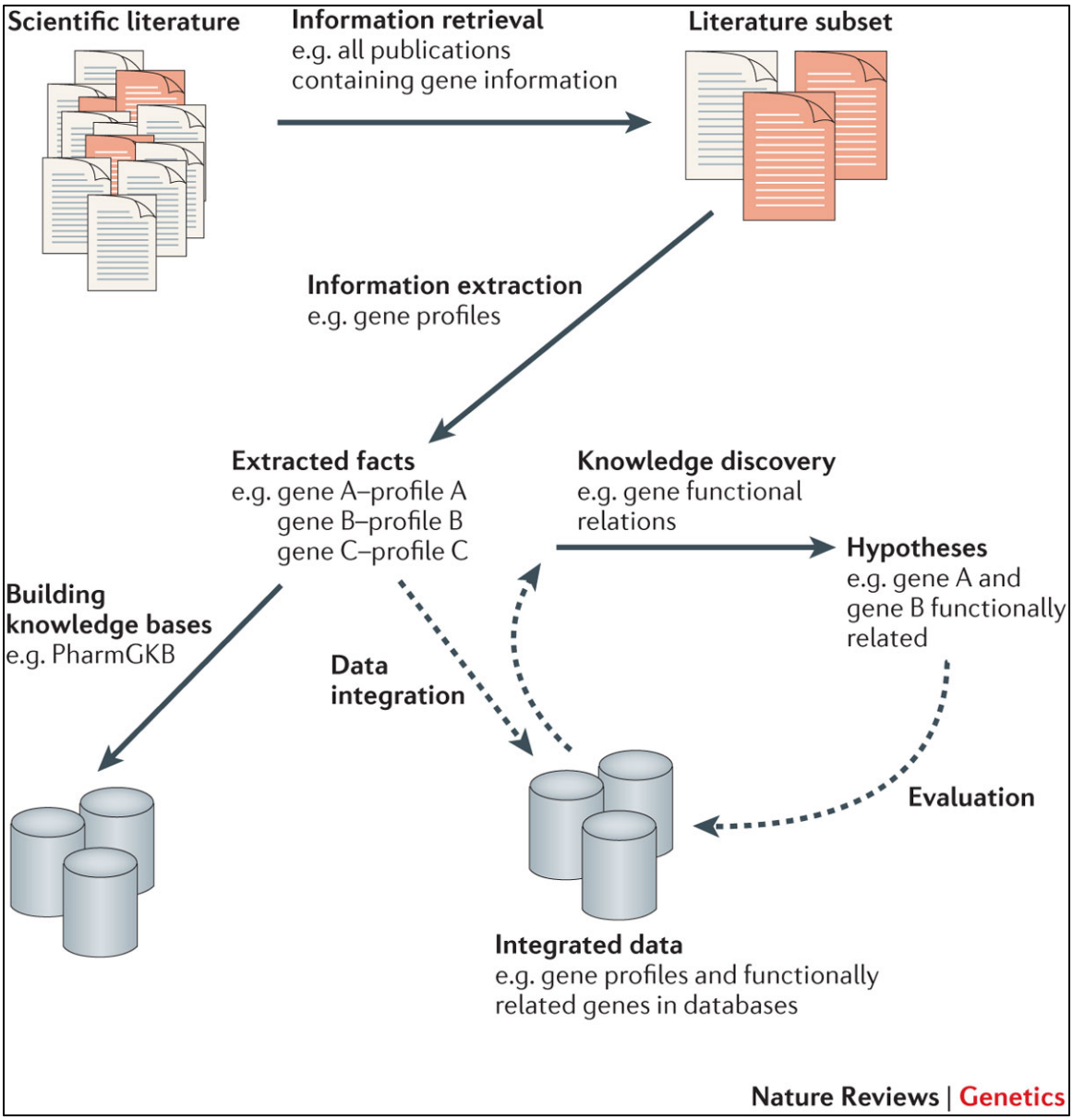

Figure 9: An overview of the different categories in which text mining can be applied. Document retrieval is the initial step and leads to the collection of documents for a given query

In one study, potential of text mining techniques have been explored to curate the HIV-1, human protein interaction database. HIV-1 is a very common pathogenic strain of virus. The database maintaining all the protein interaction in humans related to HIV-1 contains 2589 manually extracted interactions, linked to 14,312 mentions in 3090 articles. Due to advancements in text mining and to be able to rapidly extract such data from huge amounts of literature, researchers applied text-mining techniques for recreating the database. With a F-score of 88.6%, this system could recreate 50% of the interaction just from abstracts and titles, using a customized and tailored training data set and a post-processing module utilizing a dictionary with HIV and top human genes. From 49 available open-access full-text articles, this system could extract a total of 237 unique HIV-1–human interactions, whereas HHPID recorded only 187 interactions for the same articles. On an average they could retrieve 23 times more mentioned with a 6-fold increase in in unique interactions. The error analysis showed that commonly found false positive hits were due to acronyms such as cell line names or strain names. This study concludes that text-mining techniques can generate data at a faster speed, which can be used to support the manual curation process [39]. Existing approaches to mine protein-protein interactions have been broadly classified into 2 categories: Pattern matching approaches use a pre-defined set of patterns to extract protein-protein interactions [40], [41]. Parsing methods can be either shallow parsing which break sentences into non-overlapping motifs [42], [43] or deep parsing which uses the entire sentence structure and are potentially more accurate [44]. Table 3 below shows the precision and recall values for each of these methods. These are indicative figures since no benchmark data set was available to compare [38].

Category	Performance		Reference
	Recall (%)	Precision (%)	
Pattern Matching	86	94	[40]
Shallow parsing	62	89	[45]
Deep parsing	48	80	[46]

Table 3: Precision and Recall for approaches to mine protein-protein interactions

Protein interactions give a deep insight to biologists to study the mechanism of action of the living cell and ascertain potential drug targets for drug designing. Zhou et al used hidden vector state models to extract protein-protein interactions. This process of validating text mining results on protein interactions using gene expression profiles was conducted in stages: mining protein-protein interactions from literature, clustering co-expressed genes and making inferences based on the above results. Sentences were semantically parsed and trained to achieve an overall precision, recall and *F-score* of 58.3%, 76.8% and 66.3% respectively. Authors further validated these results with gene expression profiles where co-expressed gene where identified using ant base clustering technique [47].

In another study, a graph-kernel based approach has been applied extract for automated protein interaction extraction from scientific literature. Sentences were broken down using dependency parsers, and the output trees were traversed to identify sentences with interacting proteins. This method was evaluated on 5 publicly available PPI corpora and a cross-corpus evaluation was done to test whether an extraction system will work beyond

the corpus it was trained on. The method was shown to achieve performance with an F-score of 56.4 and 84.8 area under the receiver operating characteristics curve (AUC) on the AImed corpus. [48]. Authors have also attempted hybrid approaches by combining co-occurrence based approach and rule-based approach to find interactions in PubMed abstracts. They validated these extracted interactions against PPI databases and shared terms from gene ontology. According to their findings only 28% of the co-occurred pairs in PubMed abstracts appeared in any of the commonly used human PPI databases and 69% showed co-occurrence in literature [49].

These results stress on the point we made earlier that co-occurring terms cannot always make reliable protein interactions. A graph kernel based approach does not consider the biological evidence mentioned in the sentence that can strengthen the protein interaction information extraction. In a graph the distances or dependencies being considered are between protein nodes that leaves out the valuable information on interaction detection methods. To improve sensitivity of protein interaction extraction, it needs to be backed by the experimental evidences from respective articles. Articles published based on experimental studies have clearly defined section pertaining to the details of the study. Annotating biological entities with such empirical data can fasten the tasks of manual curation. As a part of BioCreAtIve III challenge, there have been studies to annotate the articles with protein interactions with the interaction detection method. In one study authors have developed a framework to identify experimental methods used to study interactions. They applied classification techniques using a combination of up to 21 features comprising regular expressions, keywords, mutual information scores unigrams

& bigrams. These features were run through different classifier like J48, Naïve Bayes ad Random Forest to achieve the highest *F-score* if 52.38% [50]. Another promising approach was to use the MeSH term ontology. Most frequent interaction detection methods were mapped to equivalent MeSH terms. Classifiers were run over pairs of text chunks and names of interaction detection methods. Vectors were built using string similarity measures like JaroWinkler [51] or TF-IDF [52]. In a study entailing use of a linear classifier using named entities as features, authors approached the task of annotating the proteins interaction documents with experimental methods not as a just a document classification task. They reported very low performance. However, they validated the results with evaluations from independent annotators [53].

All the different approaches mentioned here gave us a perspective of the task in hand. Reliability of results cannot be based upon binary classification of documents. Any sentence can form a feature that can be used to decide which category the document belongs to. However, any random textual feature cannot justify the experimental process performed to identify the interacting protein pairs mentioned in the articles. The features that should be used have to be more specific than just being from the biomedical domain. Thus we hypothesize that the experimental factors are the distinguishing factors between documents describing an interaction detection method. Documents have to be classified based on the information about experimental processes. This set the tone for our approach. We used a diverse set of experimental information as our features for machine learning techniques.

CHAPTER THREE: METHODOLOGY

A schematic overview of our methodology is shown in Figure 10. It involves information retrieval from PubMed Central. Then we extract the methods section to identify and annotate required biological entities. Using various information extraction techniques like name entity recognition, dictionary lookup and POS tagging we gather entities to build our input vector space. We classify using logistic regression with protein interaction detection methods (PIDM) as class labels.

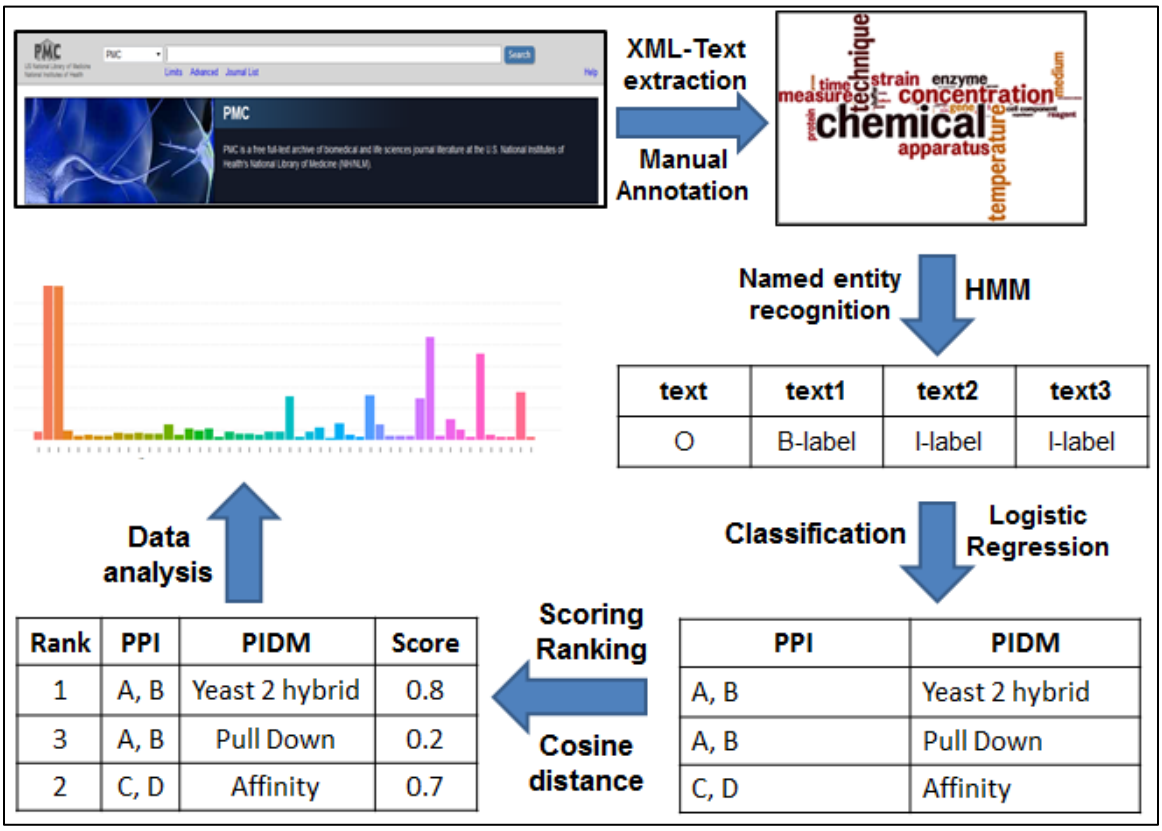

Figure 10: Methodology flowchart

Data

For classifying documents we required documents labeled with their respective interaction methods. The PSI-MI ontology [33] is the standard followed for the interaction method names used in protein interaction annotations. The PSI-MI vocabulary is rich and well controlled that explains the granularity of experimental methods used in protein interactions. BioCreAtIve used PSI-MI for data preparation for interaction

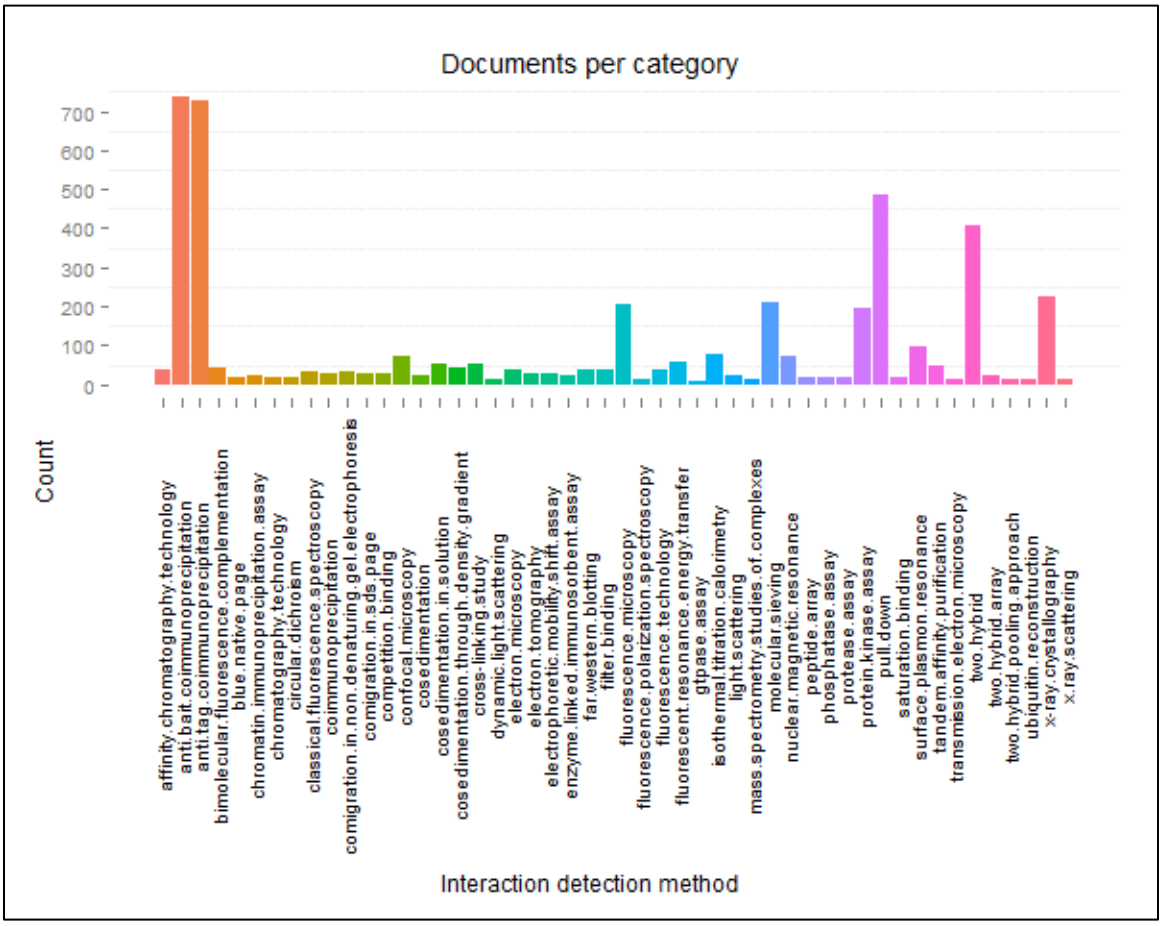

Figure 11: Documents per category

method classification task [1]. This training and the testing data were made publicly available for research. The training dataset contains 2035 full-text articles under 86 different interaction detection methods. In all it contains 4348 annotations where some of the documents overlapped under interaction methods. The test data has 305 full-text articles. Figure 11 shows the distribution of documents per category. As we can see, more that 2/3rd of the data belongs to 8 of the categories out of 86.

Approaches for feature extraction

We approached feature extraction with specificity of context in mind. The objective of classification was to categorize documents with protein interactions into respective interaction detection methods. We used biological attributes most related to the experimental methods.

a) Annotating key named entities

A wide range of keywords such as breast cancer, yeast cell cycle, metabolism etc. were used to query the full-text open-access articles in PubMed Central. Several keywords were used in the analysis so as to retrieve an extensive set of experimental factors. The queries were made using NCBI E-utilities which provides a nice interface for information retrieval from NCBI data warehouses. E-utilities stands for Entrez Programming Utilities. The E-utilities use a fixed URL syntax that translates a standard set of input parameters into the values necessary

for various NCBI software components to search for and retrieve the requested data. The Entrez system currently holds data from 38 databases covering a diverse variety of data, including nucleotide and protein sequences, gene records, three-dimensional molecular structures, and the biomedical literature. E-utilities provides interface for different languages like Perl, Python, Java and C++. A total of 8 E-utilities are available for information retrieval.

Name	Description
EInfo	Provides the number of records indexed in each field of a given database, the date of the last update of the database, and the available links from the database to other Entrez databases.
ESearch	Responds to a text query with the list of matching UIDs in a given database (for later use in ESummary, EFetch or ELink), along with the term translations of the query.
EFetch	Responds to a list of UIDs in a given database with the corresponding data records in a specified format.
EPost	Accepts a list of UIDs from a given database, stores the set on the History Server, and responds with a query key and web environment for the uploaded dataset.
ESummary	Responds to a list of UIDs from a given database with the corresponding document summaries.
ELink	Responds to a list of UIDs in a given database with either a list of related UIDs (and relevancy scores) in the same database or a list of linked UIDs in another Entrez database; checks for the existence of a specified link from a list of one or more UIDs; creates a hyperlink to the primary LinkOut provider for a specific UID and database, or lists LinkOut URLs and attributes for multiple UIDs.
EGQuery	Responds to a text query with the number of records matching the query in each Entrez database.
ESpell	Retrieves spelling suggestions for a text query in a given database.

Table 4: Description of E-utilities

Typically the process of retrieving data from NCBI using E-utilities is by searching the database of interest with certain keywords. The URL for that looks like

esearch.fcgi?db=*database*&term=*query*

This query returns IdList in XML format

```
<?xml version="1.0" ?>
<!DOCTYPE eSearchResult PUBLIC "-//NLM//DTD eSearchResult,
11 May 2002//EN"

"http://www.ncbi.nlm.nih.gov/entrez/query/DTD/eSearch_02051
1.dtd">
<eSearchResult>
<Count>255147</Count>    # total number of records matching
query
<RetMax>20</RetMax># number of UIDs returned in this XML;
default=20
<RetStart>0</RetStart>#  index of first record returned;
default=0
<QueryKey>1</QueryKey>#    QueryKey,    only    present    if
&usehistory=y
<WebEnv>0l93yIkBjmM60UBXuvBvPfBIq8-9nIsldXuMP0hhuMH-
8GjCz7F_Dz1XL6z@397033B29A81FB01_0038SID</WebEnv>
                  # WebEnv; only present if &usehistory=y
      <IdList>
<Id>229486465</Id>    # list of UIDs returned
<Id>229486321</Id>
<Id>229485738</Id>
<Id>229470359</Id>
<Id>229463047</Id>
<Id>229463037</Id>
<Id>229463022</Id>
<Id>229463019</Id>
<Id>229463007</Id>
<Id>229463002</Id>
<Id>229463000</Id>
<Id>229462974</Id>
<Id>229462961</Id>
<Id>229462956</Id>
<Id>229462921</Id>
<Id>229462905</Id>
<Id>229462899</Id>
<Id>229462873</Id>
<Id>229462863</Id>
```

```
<Id>229462862</Id>
</IdList>
<TranslationSet>          # details of how Entrez translated
the query
    <Translation>
     <From>mouse[orgn]</From>
     <To>"Mus musculus"[Organism]</To>
    </Translation>
</TranslationSet>
<TranslationStack>
   <TermSet>
    <Term>"Mus musculus"[Organism]</Term>
    <Field>Organism</Field>
    <Count>255147</Count>
    <Explode>Y</Explode>
   </TermSet>
   <OP>GROUP</OP>
</TranslationStack>
<QueryTranslation>"Mus
musculus"[Organism]</QueryTranslation>
</eSearchResult>
```

Typical pipeline for using the EUtilities that fits our bill is

```
ESearch -> EFetch
```

We then iterate over the results of ESearch to get the IDs from <IdList>. The these IDs are used in EFetch utility. The URL for EFetch looks like

```
efetch.fcgi?db=<database>&id=<uid_list>&rettype=<retrieval_type>
                  &retmode=<retrieval_mode>
```

The XPath for the for the results looks like

```
<pmc-articleset>
 <article xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:mml="http://www.w3.org/1998/Math/MathML" article-
type="research-article" xml:lang="en">
 <front>
 <body>
        <sec>
        <sec>
        <sec sec-type="conclusions">
        <sec sec-type="methods">
        <sec>
        <sec>
        <sec sec-type="supplementary-material">
 </body>
```

```
  <back>
</article>
</pmc-articleset>
```

Each section has the textual data that can be parsed using any standard XML parser. Each <sec> tag defines a particular section in the document. We were interested only in the documents that had a defined "methods" section (`<sec sec-type="methods">`). We parsed the xml documents by traversing each node to reach the methods section [54]. The HTML and XML from this section was cleaned up using boilerpipe's [55] implementation of the Boilerplate detection. Boilerplate detection algorithms help detect and remove the navigational clutter around the main textual content of a web page. It can also handle different kinds of web document formats like xml, html, json etc. Boilerplate detection uses shallow text features to distinguish between textual content and the web page navigational content. Advertisements can also be removed using these algorithms. The shallow text features considered by the algorithm are number of words, link density, element frequency, average sentence length, average word length and few other such quantitative linguistic features [56].

The extracted text was broken down into tokens using POS tagging. POS tagging is a technique in natural language processing which can assign part of speech tags to tokens per sentence. We used the HHM based POS tagging model trained on Medpost [57] corpus in LingPipe [58]. The MedPost tag set contains a list of 60 part-of-speech tags. The tags are 2-4 letter abbreviations for the part-of-speech.

For example *NN* is a "noun" and *NNP* is a "proper noun". Similarly, tags starting with *"V"* are for different kinds of verbs. MedPost also covers different kinds of punctuations, numbers and symbols. On analyzing the nouns from the broken down tokens, we hypothesized that these word features can be used to classify protein interaction detection methods. The methods' sections of 200 full text articles were manually annotated and all the nouns and noun phrases were labeled for statistical learners. Table 5 provides an example of occurrences of few experimental factors in 2 categories. The experimental factors in these articles were separated into 17 categories viz. "ANALYSIS", "ANTIBODY", "ANTIGEN", "CELL", "CELLCOMP", "ENZYME", "EXPERIMENT", "GFACTOR", "ORGN", "PHASE", "PLASMID", "PROCESS", "PROMOTER", "STRAIN", "TECHNIQUE", "TISSUE" and "TITLE".

Experimental factor/Interaction detection method	*Category*	*pull-down*	*two-hybrid*
34 °C	Temperature	0	1
Dulbeccos modified Eagles medium, fetal bovine serum	Medium	1	1
HeLa cells	Cell line	61	39
HEK-293	Cell line	76	46
Western blot analysis	Technique	47	268
30 min	Time	105	84

Table 5: Example of experimental factor features and their occurrences in detection methods

The manually annotated sentences were trained to identify named entities using the `CharLmRescoringChunker` implemented in LingPipe [58]. A rescoring chunker uses the results from the `NBestChunker` and statistics gathered from the tag transitions in the training data to re-score each of the n-best chunkings. How this works is, it uses the underlying HMM model to split sentences into syntactic structures using tokenized character sequences. For example, consider the following sentence

```
Protein|B-PROTEIN A|I-PROTEIN (|I-PROTEIN PrA|I-PROTEIN )|I-PROTEIN tagging|O (|O W303|B-STRAIN background|O )|O was|O performed|O by|O the|O PCR-based|B-TECHNIQUE method|I-TECHNIQUE (|O Aitchison|O et|O al|O .|O ,|O 1995|O )|O using|O pBXAHIS5|B-PLASMID (|O Wach|O et|O al|O .|O ,|O 1997|O )|O .|O
```

We use the IOB tag format to mark the range of the entities we want to model using HMM. In this representation, each token is tagged with one of three special chunk tags, I (inside), O (outside) or B (begin). A token is tagged as B if it marks the beginning of a chunk. Subsequent tokens within the chunk are tagged I. All other tokens are tagged O. The B and I tags are suffixed with the chunk type, e.g. B-PLASMID, I-PLASMID. Of course, it is not necessary to specify a chunk type for tokens that appear outside a chunk, so these are just labeled O. IOB tags are pretty much the standard way to represent a chunk structure [59]. The B and I tags

define the start and end position of a token. The intention of having these chunk tags is that it produces the transition states. With these chunk tags, HMM can identify all the tags that can follow a given tag or that can precede it. It calculates these state transition probabilities from the training data provided. Now, not all state transitions are legal. For example, I tag cannot follow an O tag. In such cases zero probability is emitted. When a chunker identifies a chunk tag for a token or a phrase, it is usually the one with the highest state transition probability. According to the documentation for the `HmmChunker` implementation in LingPipe [58], the number of possible transitions can be calculated using

$$numTransitions = (5 * numTypes^2) + (13 * numTypes) + 1$$

The probability of an observed output sequence $\boldsymbol{o_1, \ldots, o_{t-1}}$, produced by a state sequence say, $\boldsymbol{i_1, \ldots, i_{t-1}}$, where t is the length of the sequence can be given by

$$\boldsymbol{P(o) = \sum_i P(o|i)P(i)}$$

This mostly likely sequence of hidden states that is used to calculate the sequence of observed events is called the 'Viterbi Path' [60]. The output sequence of observed events from the Viterbi algorithm are called the first-best chunks. The rescoring process yields in a better result than any first-best chunker because it incorporates information from longer range relationships in the text [61]. For automated annotation of chemical entities we utilized open source chemistry

analysis routines implemented in OSCAR4 [24]. We used it to identify chemical names, reaction names and enzymes, if any. It provides name-to-structure parsing and vice-versa. Figure 12 shows the frequency of occurrence of the top 25 features.

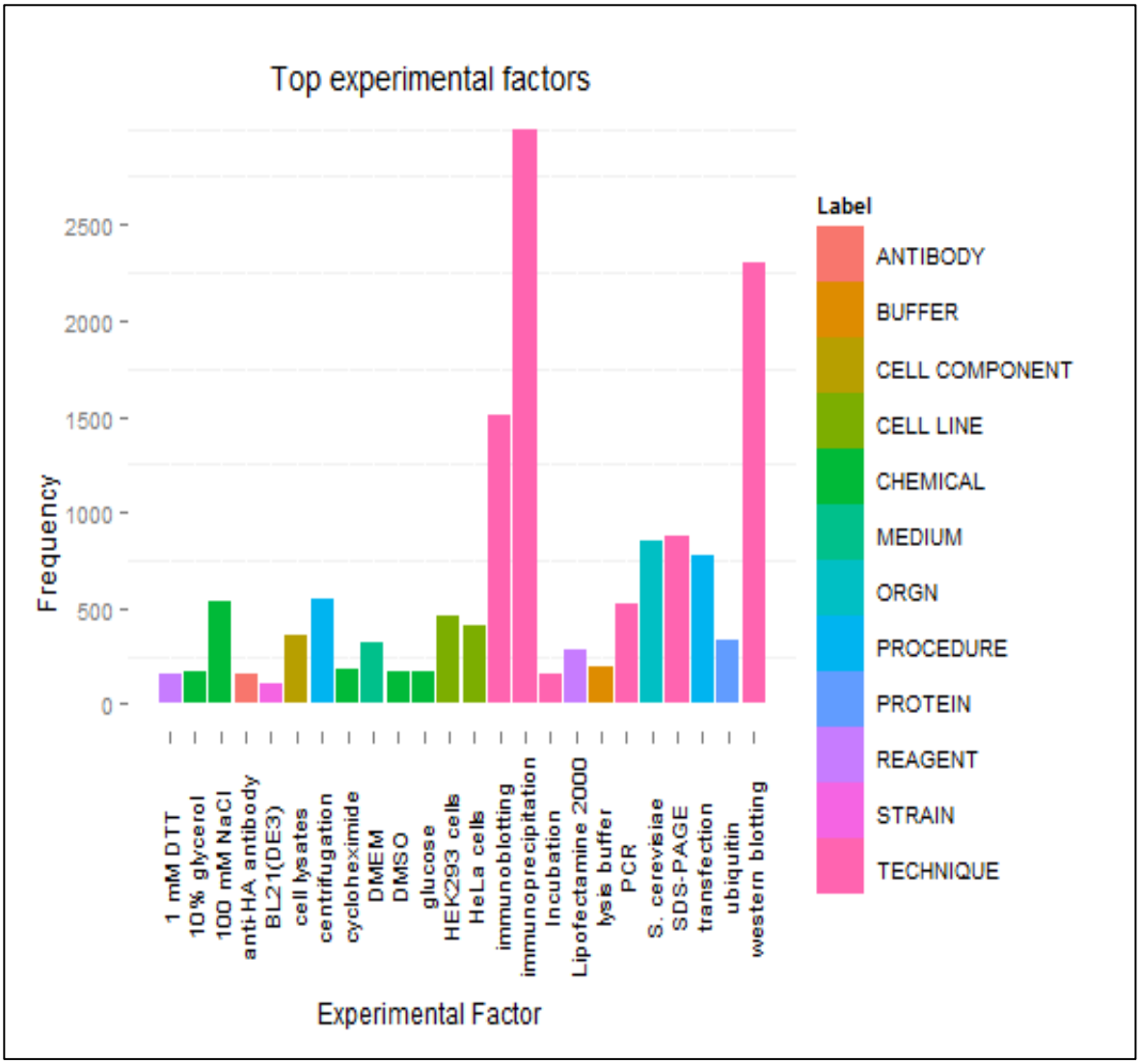

Figure 12: Top 25 occurring experimental factors

b) Experimental factor ontology look-up

Experimental attributes use many cross-domain concepts. Due to ever increasing experimental data reported as per standards like MIAME [62], an expressive ontology, experimental factor ontology (EFO), was developed. The terminology used in a particular context is restricted to a set of terms that define important aspects of a domain or application. This was the objective behind developing EFO. The scope of EFO is to support the annotation, analysis and visualization of data at EBI. EFO has a finite set of terms pulled from anatomy, diseases and chemical compounds. The EFO had 609 different types for 3889 unique terms related to experimental factors. We used these terms for a look-up in full-text protein interactions' related articles. We used the `ExactDictionaryChunker` implemented in LingPipe [58] for exact term matching against the ontology. LingPipe API was used instead of simple string matching because unlike latter it provides a nice interface for string related functions. For example, offset of the matched string and length. Also the dictionary can be serialized, for faster I/O. As mentioned before, EFO is a finite list and that is why we decided to not use a learning model and instead do a lookup. These entities are the uncommon elements (as compared to common biological entities like gene, protein) that we hypothesized to be the distinguishing factors between interaction detection methods.

c) Identifying common biological entities

Mostly commonly extracted biological entities from scientific literature are genes and proteins. Any biomedical document is bound have mentions of gene/protein entities. It's a challenge to recognize these names in text due to the ambiguity and existence of multiple synonyms for the same names. Many studies have shown efficient extraction of such mentions using different approaches like machine learning, dictionary based look-up or a hybrid approach combining the two [63–67]. A widely used corpus that has been developed to tackle this problem is the semantically annotated GENETAG corpus. It has been built from breaking down Medline abstracts into 20,000 sentences [68]. We used the GENETAG corpus in this case because the GENIA corpus was built using text for terms restricted to *human, blood cell & transcription factor.* Other reason for use of GENETAG was that is allows specific gene/protein name extraction, unlike GENIA which is generic. The corpus encompasses entities that can be categorized under proteins, DNA, RNA, viruses, lipid, cell components, atoms, body parts, cell lines, nucleotide etc., in all 36 classes. These labels were not considered in manual annotation. The common biological entities will help identify the presence of the protein/s if any, which can lead to identification of whether they interact. A confidence based rescoring chunker is used for tagging each token with one of the 36 labels.

A simple graphical representation of a document with all the feature vectors is shown in Figure 13.

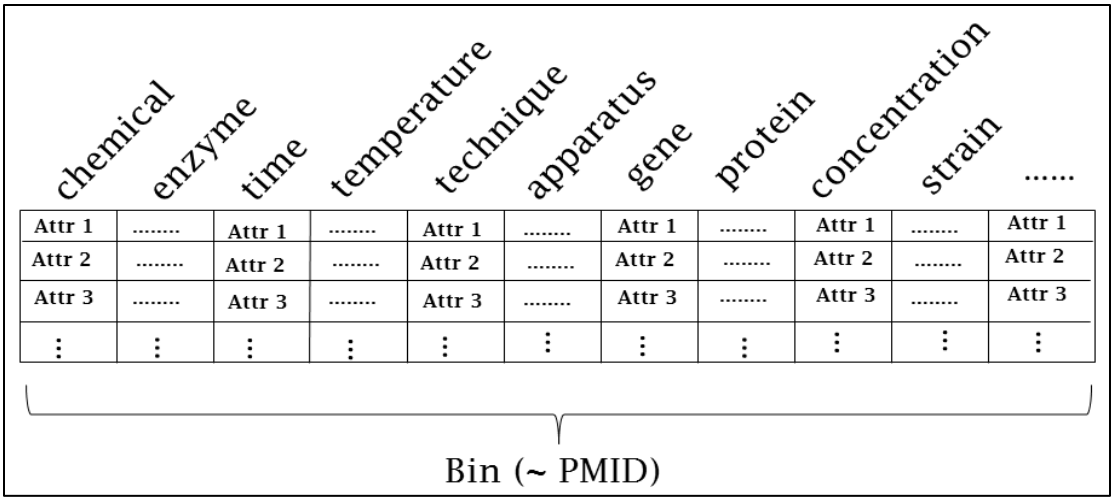

Figure 13: Graphical representation of document as set of feature vectors

Classification

Classification is a technique of assigning a new input vector to one or more of pre-defined categories. A few set of quantifiable properties and features are analyzed for each input. These features can be ordinal, integer, real values etc. Any algorithm that implements classification and uses such features is a classifier. There are many classifiers suited for different problems. To name a few, Naïve Bayes, SVM, Logistic Regression, Decision Trees, Multi layer perceptron, J48, Random Forest. In this study, we chose to utilize the Logistic Regression Classifier. Logistic regression is one of the best discriminative probabilistic classifiers, measured in both log loss and first-best classification accuracy across a number of tasks. Using this method, we classified protein

interaction documents into one or more interaction detection methods. It is not possible to sift through all the possible combinations of parameters obtained from the documents. The strength of using logistic regression for our data lies in its interpretability of the parameter estimates. Logistic regression does not impose any restriction against them being correlated [69].

Logistic regression is a technique to learn functions of the form $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ or $\boldsymbol{P}\ (\boldsymbol{Y}|\boldsymbol{X})$. If there are $\boldsymbol{k}$ categories, the model will consist of $\boldsymbol{k-1}$ vectors $\boldsymbol{\beta}[\boldsymbol{0}], \ldots, \boldsymbol{\beta}[\boldsymbol{k-2}]$. Then for a given input text vector $\boldsymbol{x}$ of dimensionality $\boldsymbol{k}$, the conditional probability of a category given the input is defined to be:

$$\boldsymbol{P}(\boldsymbol{c}j\boldsymbol{x}) = \frac{\boldsymbol{exp}(\boldsymbol{\beta}[\boldsymbol{c}] * \boldsymbol{x})}{\boldsymbol{1} + \sum_{\boldsymbol{i}<(\boldsymbol{k}-\boldsymbol{1})} \boldsymbol{exp}(\boldsymbol{\beta}[\boldsymbol{i}] * \boldsymbol{x})}$$

For evaluation we used F-measure, which is calculated using Precision and Recall. In natural language processing and text mining, a new technique can be compared to a current technique, using a test data to estimate if it produces a better metrics. Assuming the null hypothesis, that is the new technique is similar to the old technique, what is the probability that the results produced on test data set would be skewed in favor of the new technique. If this probability is less than at least 5%, one can reject the null hypothesis and conclude that differences in the results are statistically relevant at that threshold level of 5% or less. Precision, recall and a balanced F-score are some methods that can help deduce this statistical significance [70]. In other words, a good relevant hit is useful to the user, but a bad irrelevant hit can be time consuming and cost ineffective. Therefore, measure of quality is based on relevance of documents returned by the system. The

topmost documents are especially important since the user will mostly read them. A non-relevant document at top of the list will thus have a higher cost associated with it [71].

Precision is the probability that a randomly retrieved document is relevant i.e. how well the system performs in not returning non-relevant documents [71], [72]. It is also known as the positive predictive value, which is defined as the proportion of positive test results that are true positives. Precision takes all documents into account, but a threshold can be set that considers only topmost results returned by the mining system. This is called precision at n. [72]

Mathematically, Precision can be defined as [71],

$$Precision\ (P) = \frac{\text{Number of Relevant Documents Retrieved (X)}}{\text{Total Number of Retrieved Documents (Y)}}$$

Recall is the probability that a randomly selected, relevant document is retrieved in the search results i.e. how well the system performs in finding relevant documents [71], [72]. A 100% recall value can be achieved by a system returning all the relevant documents from the entire collection. Therefore, recall by itself is not a good measure of quality of the system [71].

Mathematically, Recall can be defined as [71],

$$Recall\ (R) = \frac{Number\ of\ Relevant\ Documents\ Retrieved\ (X)}{\text{Total Relevant Documents in Collection (Z)}}$$

Recall and precision are inversely related. As precision goes up, recall goes down and vice versa. This relationship depends on the language used for retrieval. If the systems combines Boolean (to include synonyms, related terms, general terms, etc.) rather than proximity operators, precision will suffer because synonyms may not be exact synonyms, and irrelevant document retrieval increases. Unfortunately if the system does not use these Boolean operators, it will not achieve high recall [73].

F-measure or F1 score is the measure of a test's accuracy, which takes the harmonic mean of precision and recall into consideration [72].

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Precision and recall are evenly weighted in the above formula.

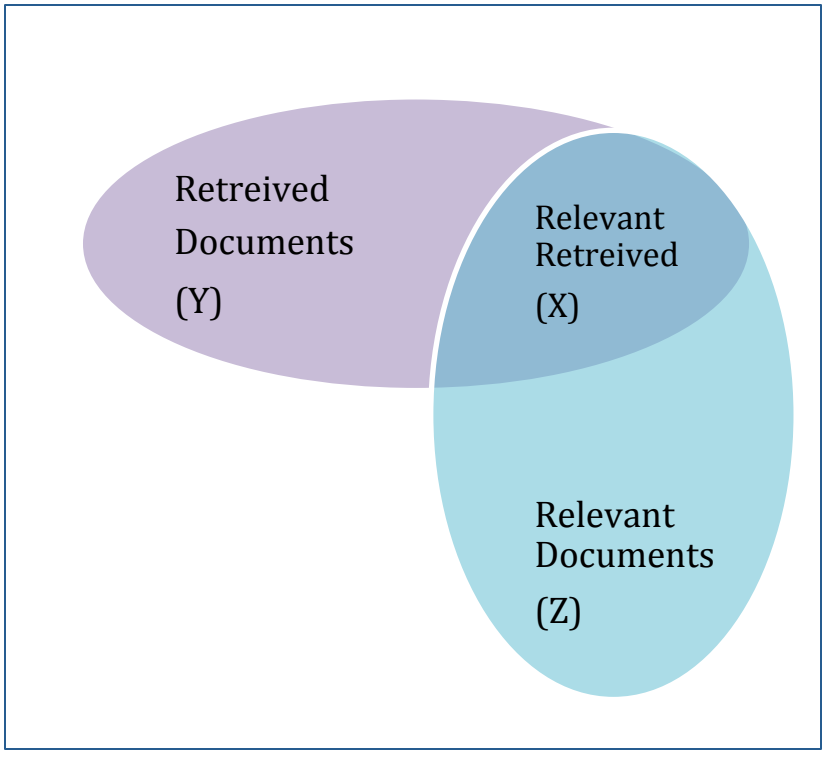

Figure 14: Visual representation of evaluation method statistics

Accuracy of a system can be calculated by

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

The accuracy of a system can be defined as the ratio of closeness in measurement of a certain quantity to the quantity's true value [74]. The accuracy, in general, is a measure of how well a system can perform.

Generalization of learning of a process of applying results from a sample and applying it to a population. This requires the sample to be randomly selected and to be a representative of the population. A fundamental point behind generalization is that statistical numbers depend on the process by which they were derived [75]. In simple words, for a system to perform well on an unknown test dataset, it has to perform with 100% accuracy (in theory) on the known train dataset. Only then can a system be said to have generalized. When a randomly selected sample, with features representative of the whole dataset, is tested using such a system the outcome can be relied upon.

Extraction of protein interaction sentences with experimental evidences

The main aim for our study has been to categorize documents containing protein interactions into relevant interaction methods. This will greatly aid in the process of manual curation, where experts do not have to go through the full article to be able to get a perspective of what interaction detection method is discussed in the article. To make it even easier for extracting complete protein interaction information, we have attempted to retrieve the sentences with mentions of interacting proteins along with the interaction detection method. We are using Lucene [76] and its full-text indexing capabilities along

with a set of custom rules for acceptable results. A document is searched for sentence-based evidence to validate the occurrence and extraction of protein interaction. The outcome of classification of document into interaction detection method is used along with certain rules to search. A Boolean search query, for example would be

$$interaction\_detection\_method\ AND\ num\_proteins\ >\ 2\ AND\ interaction\_verb$$

The score of query for document correlates to the cosine-distance or dot product between document and query vectors. A document whose vector is closer to the query vector is scored higher. Precision is the percentage of relevant documents among all the retrieved documents. Recall is the percentage of relevant documents that are retrieved in response to a query. Precision and recall cannot be regarded as measure of effective retrieval. However, the effort required to sort out all the relevant documents from the retrieved set of documents could be pretty high. The use of and operator, like in the above query, narrows the search, thus improving precision at the cost of recall [77].

Evaluation against BioGRID

BioGRID only annotates data that is supported with experimental evidence in scientific literature. It annotates two kinds of interactions: protein and genetic. BioGRID does not directly annotate using the PSI-MI standards. However, a large set of annotations can be directly mapped to the PSI-MI vocabulary [1]. We validated our algorithm against a test

dataset and also a small BioGRID dataset. Number of entries in BioGRID that map to PSI-MI ontology:

259256	MI:0254 (genetic interference)
208238	**MI:0004 (affinity chromatography technology)**
78948	**MI:0018 (two hybrid)**
29667	**MI:0096 (pull down)**
19381	MI:0401
12289	MI:0415 (enzymatic study)
7473	MI:0090
2227	MI:0686
2116	MI:0428 (imaging technique)
1711	MI:0114 (x-ray crystallography)
822	MI:0047 (far western blotting)
623	MI:0055 (fluorescent resonance energy transfer)

Table 6: BioGRID entries mapping to PSI-MI ontology

We selected 75 documents from MI:0004, MI:0018 and MI:0096 for further validating our approach. The reason we selected these interaction detection methods was that it had full-text articles available in our test data.

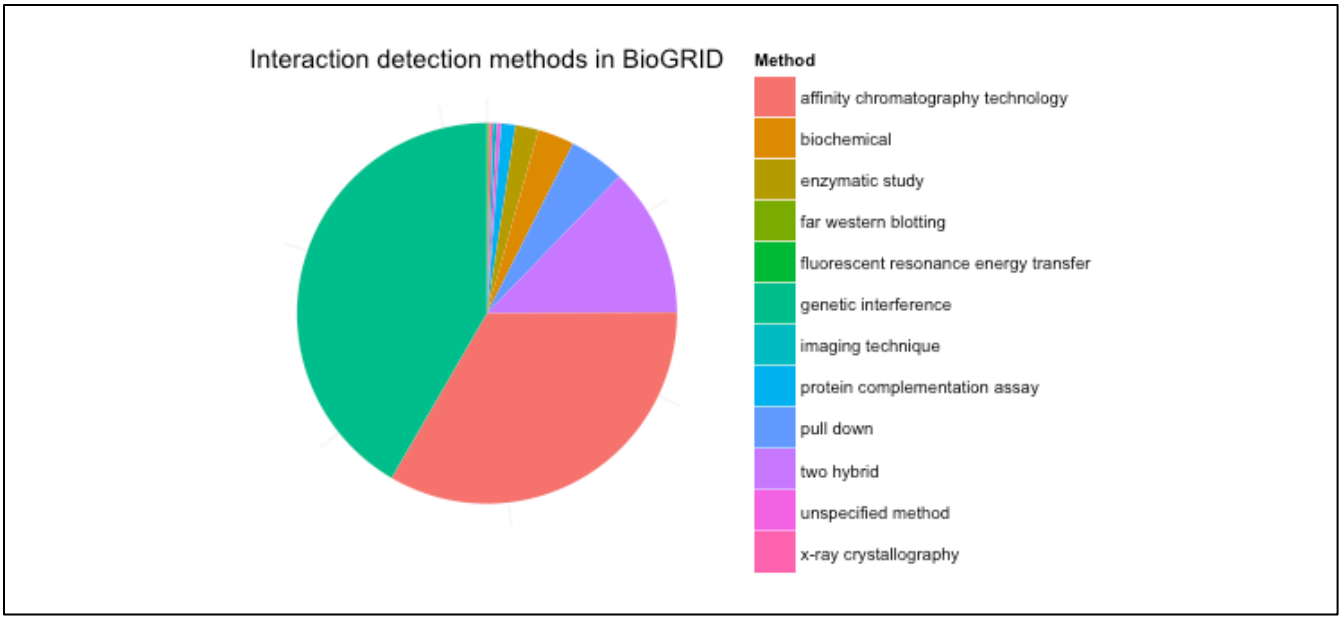

Figure 15: Distribution of interaction detection methods in BioGRID data

Other than the top 25 detection methods, the rest did not have enough full-text articles associated with it in the training data. The less popular methods had only 1-10 articles falling under its category. As we are approaching the problem with context specificity, less training data would mean less context specific features. Hence, we chose to proceed with the top 25 experimental methods, which contributed to almost 90% of the training data.

Evaluation of generalization in learning

Before testing the performance of the systems on the test data, we tried evaluating the performance against the train data itself. This is done to understand if the system generalizes. We trained a total of 4598 documents and tested on the same 4598 documents. The system performed with an accuracy of 93.68% and a maximum F-measure of 59.6%. It showed a very high specificity of 96.7%, which indicates low, type I error rate.

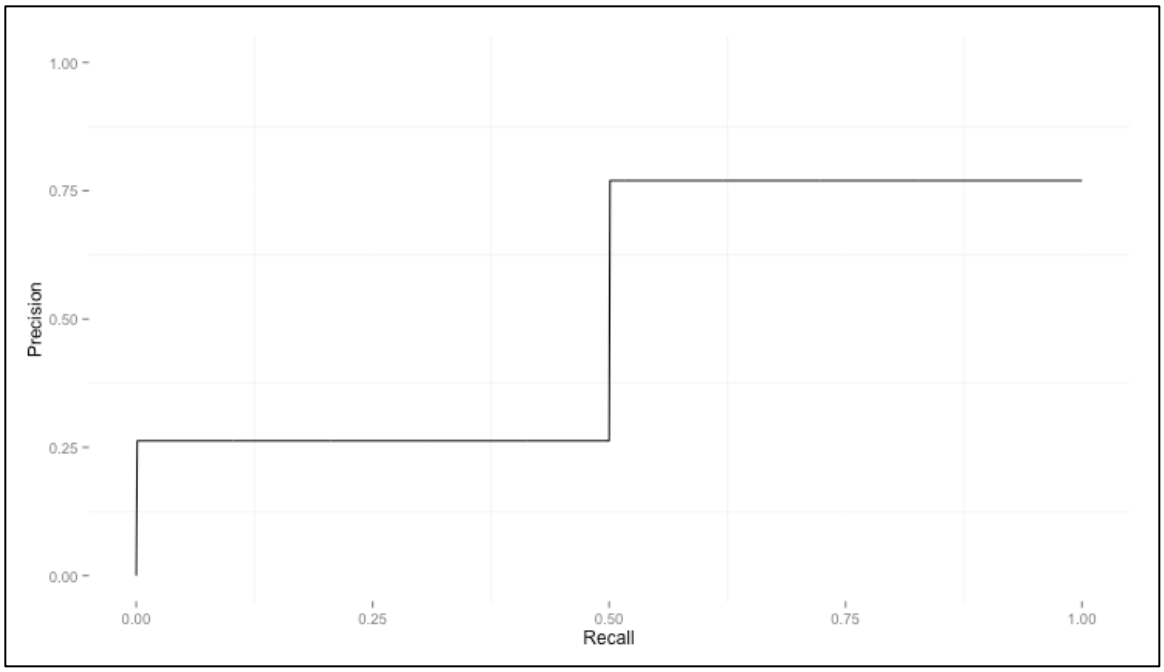

Figure 16: PR curve for evaluating generalization of system. Training and testing was performed using the same dataset

Evaluation against test data

For testing on gold standard data that had 300 documents, we performed multiple runs of logistic regression classifier by slightly tweaking the training data. The results in Table 7 are average values from multiple runs for each type. We also performed 10-fold cross validation on each run. Also we randomly selected features from annotated entities, common biological attributes or experimental factors. For the runs 2, 3 the combinations of all feature performed the best for us. For balancing the data, we randomly selected similar numbers of articles in the training data for each experimental method. In the same run, we also increased the data in the order 20%, 40%, 60%, 80% and then the whole set. This run performed the best for us with a F-measure of 47.6%. The other run we

performed was using the Top 6 experimental methods MI:0006, MI:0007, MI:0096, MI:0018, MI:0114, MI:0071. We resampled the data with every run. The overall performance for this run showed a F-measure of 45.2%. Resampling can help with altering the original class distribution. This further results in distortion of output of the classifier. To recalibrate the output every time to match the original class distribution is a challenging task. In our case, the lack of such recalibration may have resulted in the over-estimation of probability of the minority class. We also observed that the best performance was achieved when a minimum of 250 features was used.

#	Data	Accuracy	F-Measure
1	All data	0.80633	0.386
2	Balanced data	0.75108	**0.476**
3	Top 6	0.7127	0.452

Table 7: Classification performance for multiple runs on test data

Detailed results of each individual run are tabulated below

Accuracy	F-Measure	Comment on data
0.75108	0.475675676	With half data of MI:0006, MI:0007, MI:0096, MI:0018, MI:0004, MI:0114
0.90978	0.404278846	Without MI:0006, MI:0007, MI:0096, MI:0018, MI:0004, MI:0114
0.80633	0.38585034	8 categories, minFeature=250, numFolds=1000

0.80633	0.38585034	8 categories, minFeature=750, numFolds=1000
0.92734	0.323575949	Without MI:0006, MI:0007, MI:0096
0.75696	0.318196721	With MI:0006, MI:0007, MI:0096, MI:0018, MI:0004, MI:0114
0.93459	0.299576271	All data, minFeature=250, numFolds=1000
0.926	0.290890585	Without MI:0018, MI:0004, MI:0096, MI:0114 (BioGRID)
0.80562	0.269704433	8 categories with more than 150 documents
0.80562	0.269704433	8 categories with feature set 2
0.92982	0.235658915	Half data in MI:0018, MI:0006, MI:0007, MI:0096
0.933461538	0.197631579	All data, minImprovement=0.00001
0.9306	0.176581197	With all 25 categories

Table 8: Detailed results of classification run by sampling data

We chose to use the Precision-Recall evaluation methodology, which can be combined in a single quality measure, the F-measure, as reported in Table 7. Precision quantifies the amount of noise in the output of a detector, while Recall quantifies the amount of ground-truth detected. A summary statistic for the performance of a classifier is reported by maximal F-measure on the PR curve. Figure 17 shows a precision-recall curve with un-

interpolated data. Although the curve is in the lower left portion of the graph, indicating low overall performance of the classifier with an area under curve of 31.44%, it has shown a maximal F-measure of 61.8%. Maximal F-measure is the highest on the surface of the curve. Maximum F-measure is, as mentioned before an overall summary statistic. And the F-measure in Table 8 is the micro-average value calculated from the summation

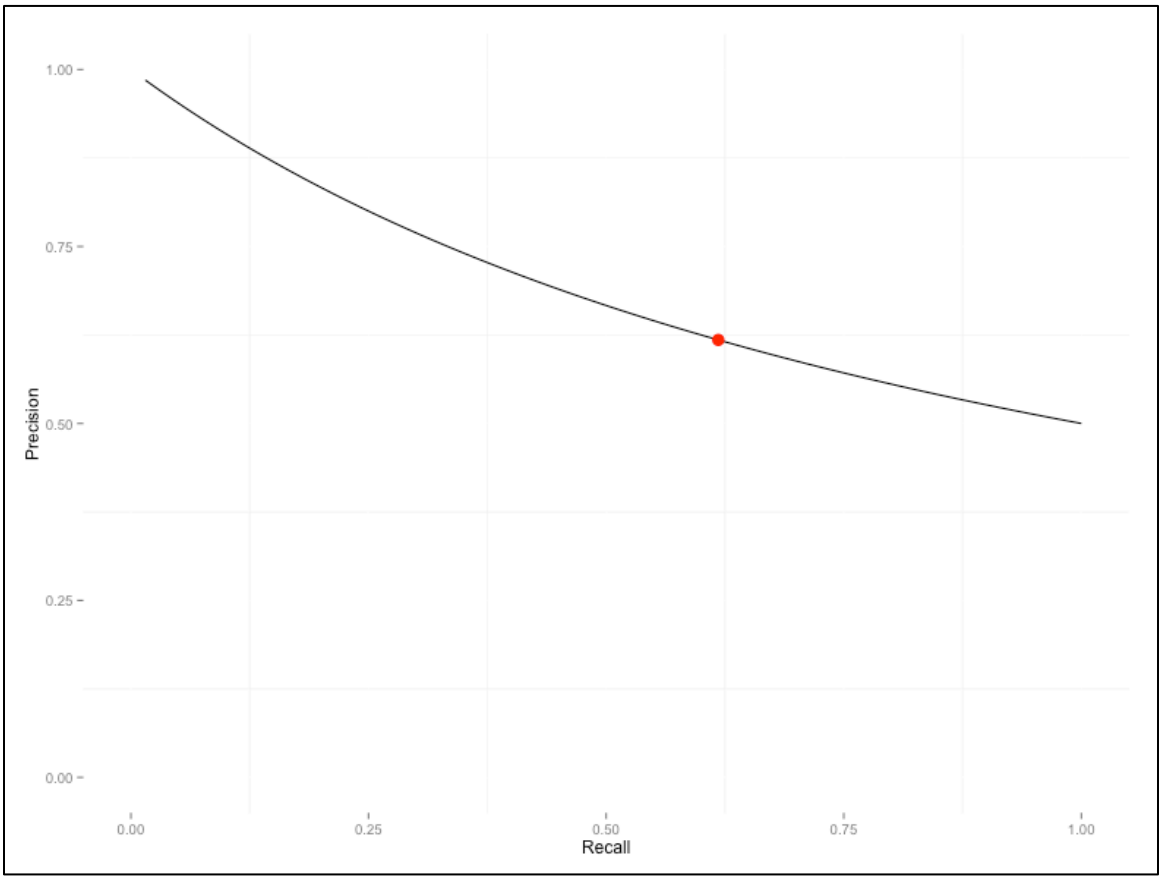

Figure 17: Precision-Recall curve with maximal F-measure

of individual TP, FP and FN for each run for the particular set of data.

For further evaluation of the system, we trained the system with the top five categories which had the most number of training documents and then tested it against the whole test set. The system will be said to be performing well, if it can identify most of the true

negatives, as the test data will have categories that were not used to train the classifier. For training we used MI:0006 (anti-bait coimmunoprecipitation), MI:0007 (anti-tag coimmunoprecipitation), MI:0018 (two hybrid), MI:0096 (pull down), MI:0114 (x-ray crystallography). We trained a total of 2572 documents under these categories. On testing it against the complete set of 300 documents in 25 categories. It could identify total of 60 positive references and 469 negative ones. The total number of cases is 529 due to the overlap of documents under certain categories. When tested against the same five categories that it was trained against it identified 60 positive references and 207 negative references. It indicates that the classifier did not fail to identify the true positives. However, it could also identify the true negatives thus keeping a low type I error rate.

Some of the top 5 features for the top 5 interaction detection methods are listed below

#	**anti-bait coimmunoprecipitation**	**anti-tag communoprecipitation**	**two hybrid**	**pull down**	**x-ray crystallography**
1	immunoprecipitation	immonuprecipitation	PCR	GST	Crystal structure
2	transfection	transfection	Cytoplasm	deletion	Monomer residues
3	NaCl	deletion	GST	EDTA	Hydrogen bonds
4	phosphorylation	nucleus	Mammalian cells	light	Electron density
5	Immunoblotting	Western blotting	*Saccharomyces cerevisiae*	*E. coli*	temperature

Table 9: Top 5 features for top 5 categories

We validated the results of classification against BioGRID by extracting experimental evidence based sentences for protein interactions. Among all the PubMed references used in BioGRID, 75 were available as full-text in our test data. Out of the 75, 7 articles belonged to MI:0018, 23 belonged to MI:0096 and 45 belonged to MI:0004. We observed that just one article with PMID 20508643 under the category MI:0004 had reported 757 interactions. However, there was not even a single mention of the word "affinity chromatography technology" in the document. Neither were there any mentions of synonyms for MI:0004, which are 'affinity purification' and 'affinity chrom'. The aim of our study is to annotate mentioned of protein interactions with experimental evidence. As there were no mentions of the method used in most of the articles associated with 'affinity chromatography technology', we proceeded with the articles falling under other 2 categories, MI:0018 and MI:0096. These 2 interaction methods had mentions of 64 interactions across 30 full-text articles. 3 of these articles over-lapped between the 2 methods. The classification using only these two methods for training and testing performed well with a F-measure of 69.91%. We indexed all the unique 27 documents using Lucene [76]. We processed 32,205 sentences in all to identify 39 sentences that match our Boolean search pattern. Amongst these 39, 20 had mentions of protein interactions along with an experimental method giving us a precision of 51.28% and a low recall of 31.25%. We manually verified these interactions correctness of experimental evidence. Some examples of relevant search results retrieved

```
PMID: 18687693
Sentence #: 286
Sentence: DISCUSSION Herein, using the two-hybrid system,
we
     identified DP-1 as a SOCS-3-interacting protein.
IMT: two hybrid
```

Proteins: DP-1, SOCS-3

PMID: 18337465
Sentence #: 202
Sentence: (B) By yeast two-hybrid assays, the UNC-89 PK2 region
interacts with SCPL-1, but not SCPL-2, -3, or -4. UNC-89 PK2|SCPL-1
IMT: two hybrid
Proteins: UNC-89 PK2, SCPL-1

PMID: 19941825
Sentence #: 878
Sentence: Rep Interacts with DnaB (A) Binding of Rep and UvrD to
surface-immobilized E. coli and B. stearothermophilus DnaB (860 and 1705 resonance units, respectively), as measured by surface plasmon resonance.
IMT: surface plasmon resonance
Proteins: DnaB (A, Rep, UvrD,

PMID: 19747491
Sentence #: 369
Sentence: Surface plasmon resonance analysis SPR analyses of
RelB—RelE and RelB—Lon interactions were carried out on a Biacore 3000 instrument (Biacore AB) equipped with a CM5 sensor chip.
IMT: surface plasmon resonance
Proteins: RelB, RelE, RelB, CM5 sensor chip,

PMID: 18945678
Sentence #: 289
Sentence: Confocal fluorescence microscopy of HEK293 cells
transiently expressing either myosin RLC fused to YFP or NR2A-(1—1028) fused to CFP (hereafter known as NR2A) revealed a predominant intracellular distribution of both proteins that could be clearly distinguished from that of YFP or CFP alone (Fig. 6, compare YFP alone in A—C with D—F and G to H; CFP alone is not shown).
IMT: fluorescence microscopy
Proteins: myosin RLC, YFP or NR2A-(1, NR2A, YFP, Fig. 6, C,

PMID: 18713736
Sentence #: 178
Sentence: D, fluorescence microscopy of CHO pgsA-745 cells
transfected with a mouse GPIHBP1 expression vector,
revealing that the binding of DiI-labeled chylomicrons
to GPIHBP1 can be blocked with immunopurified
antibodies against the acidic domain of GPIHBP1.
IMT: fluorescence microscopy
Proteins: GPIHBP1, immunopurified antibodies, acidic
domain, GPIHBP1,

PMID: 18682389
Sentence #: 296
Sentence: CT-Mlp1 and Mlp1-NBD were expressed in yeast
cells
expressing ΔRGG-Nab2-GFP, which displays localization
throughout the cell (36), and ΔRGG-Nab2-GFP was
visualized by direct fluorescence microscopy.
IMT: fluorescence microscopy
Proteins: CT-Mlp1, Mlp1-NBD, RGG-Nab2-GFP,

PMID: 19088068
Sentence ID: 254
Sentence: SDS-PAGE (14% gel) and Western blotting analysis
of
H2AZ pull-down by SWR1(1— 681) or SWR1(⌬N2) complexes
at the 0.2 or 0.3 M KCl condition.
IMT: pull down
Proteins: SWR1, SWR1, N2) complexes,

PMID: 19088068
Sentence ID: 334
Sentence: Recent studies have shown that an N-terminal
subdomain
(residues 340 — 411) of Swr1, the HSA domain, is
sufficient to pull-down Arp4 and Act1 (42), and thus
can be considered as a binding platform for Arp4 and
actin.
IMT: pull down
Proteins: N-terminal subdomain (residues 340 — 411, Swr1,
HSA domain, Act1, Arp4,

A graphical representation of the evidence extraction, using search is shown in Figure 18.

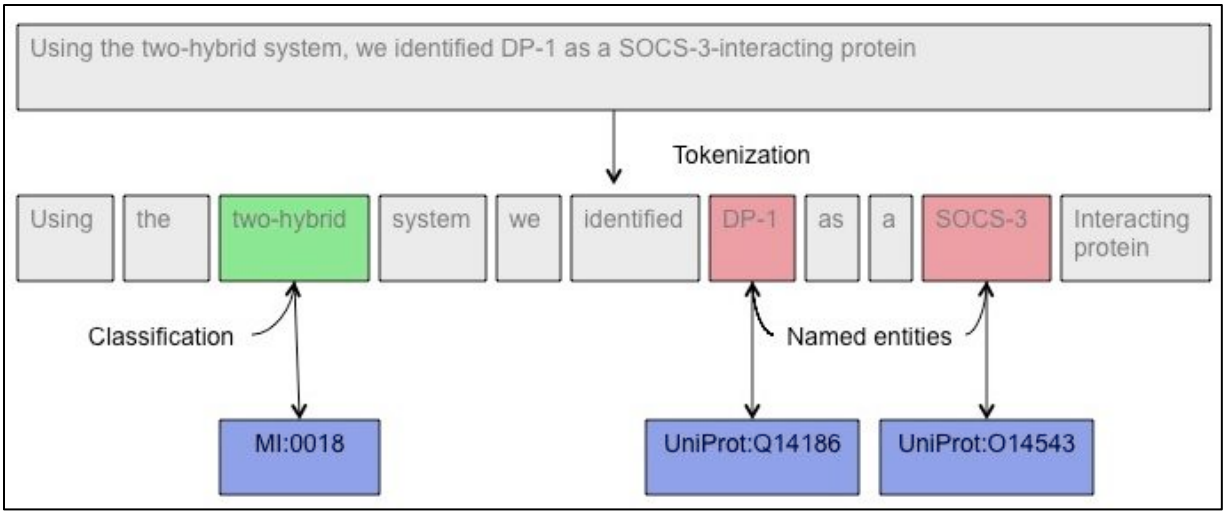

Figure 18: Graphical representation of search results for evidence extraction.

PMID	Interactor A	Interactor B	Interaction detection method
18337465	UNC-89	SCPL-1	two hybrid
18337465	UNC-89 PK2	SCPL-1	two hybrid
18337465	UNC-89 PK2	SCPL-1	two hybrid
18337465	SCPL-1	Fn3-Ig-PK1	two hybrid
18337465	protein phosphatase	UNC-89	two hybrid
18687693	SOCS-3	DP-1	two hybrid
19112176	TDP-43	UBQLN	two hybrid
19112176	Polyubiquitylated TDP-43	UBQLN	two hybrid
19164295	TRF1	RLIM	two hybrid
18840606	MLL1	GST	pull down
18840606	GST-MLL	WDR5	pull down
19088068	HSA domain	Arp4 and actin	pull down
19088068	N-terminal Region of Swr1	H2AZ	pull down
19112176	TDP-43	UBQLN	pull down
19561358	anti-His6 antibody	FANCD2	pull down
20200159	Tup12	Lkh1	pull down
20200159	Lkh1	Tup11 and Tup12	pull down
20200159	Lkh1 ED665H cells containing pESP	Tup11 and Tup12	pull down
20339350	GST-USP9x	HisEFA6A	pull down
20407420	TOC1	PRR5	pull down

Table 10: Protein interaction mentions with experimental evidence that are present in BioGRID

CHAPTER FIVE: DISCUSSION

With this study we approach the task of annotating protein interactions with experimental data using context specific information. For example, in the context of protein interactions and experimental factors the most important biological attributes that can serve as features for text mining techniques are the experimental factors, and biological named entities. Feature extraction is done from a section of document that mostly explains setup of the study. Unlike conventional techniques, any possible feature from the text is not utilized. Biological outcome can vary depending on a lot of factors. For example, a peptide bond can break in the presence of water. Such a text mining approach can control the reliability of the results, in case of biological data. Though the performance of our system is not very high, it can still bring up evidences of experimental methods for annotating protein interactions. One of the reasons for the low performance of our system is the availability of data. Not many of the PubMed references from BioGRID are publicly available through PubMed Central OAI (open access). This approach is heavily dependent on accessible full-text documents. The high imbalance of data for certain experimental methods can cause statistical learners to over-estimate probability for less occurring classes. On the other hand, with well-balanced data, our system showed pretty good performance. Another concern with information extraction, like in this case, is the format of data. PDF files are not the easiest to parse. On conversion to plain text, the layout is found to be broken, sentences are incomplete, and

symbols cannot be represented in ASCII etc. This again stresses on the fact that there needs to be an easily readable open document format.

Running our approach is realistic as compared to the curation effort that is put behind each document. With the utilization of such an approach, manual curators can get a perspective of what experimental method the document is about. They will also have the protein interaction sentences with experimental evidence. The approach justifies within an effort to speed-up the process of curation, with a long-term goal of annotating and linking all the possible literature there is. The task is more than classifying only the textual features. It is about classifying features pertaining to the given context. We observed that features from all the defined categories were covered, however the techniques, strains and medium were ranked higher. Also we observed that the documents related to 'affinity chromatography technology' that we obtained from the test data that have been annotated on BioGRID [3] hardly had the mention of the words: 'affinity' and 'chromatography'. So even if the document is correctly classified as being related to affinity chromatography, we could not pull up sentence based evidences for protein interactions.

We used the PR curve evaluation method over the ROC curve evaluation. They key difference between ROC and a PR curve is that the ROC curve will be the same no matter what the base probability is. This makes the PR curve a little more useful. A ROC curve is made up of recall/sensitivity and specificity, which are both probabilities conditioned on true class label. Precision is a probability conditioned on your estimate of

the class label and will thus vary if you try your classifier on multiple classes. This difference may not matter in case of a binary classification like, for example, is the given token a gene or not. This is because, in such a case a question like "What is the probability that the token is a gene given my classifier say it is?" is directly answered. In this study the expected out for our classifier is not just predicting if the document is related to a interaction detection method, but to correctly assign a class label to the document. Thus, a PR curve evaluation fits our scope better.

An argument that can be put forth for such an approach is why use classification and then search using the results of prediction. It is, in our opinion, a valid point. However, the problem at hand is not just extracting sentences with mentions of interacting proteins and experimental methods. The challenge is to condense the document pertaining to a study mainly involving protein-protein interactions identified using a certain interaction detection method. For example, consider the data mentioned in Table 5. A simple search using queries like "two hybrid" or "pull down" might return all the sentences with its mention. However, that does not justify if the protein interactions in the document were identified using either "two hybrid" or "pull down". If we look at the contents of the document we can se that it may contain; 34 °C (1), Dulbeccos modified Eagles medium, fetal bovine serum (2), HeLa cells (3), HEK-293 (4), Western blot analysis (5), 30 min (6). With utilization of classification techniques, if we come across a feature vector ($feature\_id: feature\_value$) like $1:1, 2:1, 3:39, 4:46, 5:268, 6:84$, we can, with certain confidence, say that the document is about "two hybrid" interaction detection

method. Similarly if the feature vector looks like $1{:}\,0\ 2{:}\,1, 3{:}\,61, 4{:}\,76, 5{:}\,47, 6{:}\,175$, we can say the document is about the method “pull down”.

Following is an enumeration of some of the critical issues of the system with the feasible solutions

1. One of the major issues is due to unclean data. The data available is converted from PDF format to text. PDF to text conversion has a few drawbacks. One drawback is non-ASCII to ASCII conversion of characters. On conversion of PDF to text, the output is found to have broken layout, incomplete sentences and as mentioned before broken conversion of non-ASCII characters. This adds a lot of noise.

- A feasible solution would be availability of text in a plain text clean readable format. If the document is converted to plain text from PDF the algorithms have to be better at handling character encoding.

2. With PDFs or text documents, the different sections of a document cannot be identified. The introduction or the methods section is not distinguishable from the supplementary material. Even a mention of some of the feature tokens goes through the training and testing process.

- An open and clearly defined document format can be a solution to this problem. The XML format provided by PubMed Central has clearly defined sections in the document, which can avoid unnecessary consumption of data from rest of the document.

3. The training data is very unevenly distributed in each of the categories. Some categories have around 700 documents in training, while some other have around 10-30. The high imbalance of data for certain experimental methods can cause statistical learners to over-estimate probability for less occurring classes.

- One way to improve this issue is to add more data. However, due to less usage of some of the categories, there are not sufficient published articles available for those categories. Another approach to solve such a problem would be to use data resampling or extrapolation

We used LingPipe [58] for most of the implementations that were required for the study. LingPipe is a very well implemented and documented software package. It is widely used in the academia, as the source if freely available. The tool also has over 50 citations in published articles. An important feature is it provides trained models on popularly distributed biomedical corpus to named entity recognition [78]. This makes it a tested and evaluated tool for research purposes. LingPipe provides well-implemented and intuitive interfaces to some of the common text manipulation tasks. With about 4000 documents distributed into 25 different categories for training the logistic regression classifier, the task was computationally exhaustive. The number of feature vectors is calculated by $numFeatureVectors = numCategories - 1$. Our feature vectors were dense as they were composed of 200+ features. Moreover, due to overlap of certain documents between interaction detection methods the feature vectors were correlated. This is also one of the reasons we chose a logistic regression classifier. However, this resulted in a computationally exhaustive training process, which also made it very time consuming.

For running 10 documents under 3 categories, with feature extraction and probability estimation, the process took about 45 minutes. Thus we broke down the problem by caching the features from training and testing data. We used file-based cache for features of each full-text document. The PubMed ID was used as the cache identifier. For training and testing we just loaded the features for a particular document identified by it PMID from the respective cache files. This reduces the task of feature extraction at runtime, and thus reduces he consumption of system memory. Also, whenever a feature set for a document is required, it is available for look-up using the PMID. Thus, it is not required to save all the features in the memory all the time. Another reason to use LingPipe has been its implementation in Java. Java is a beautifully designed, to have few implementation dependencies. It is concurrent and an object oriented language. It is well backed up by Oracle and the open source community. It can run on any Java Virtual Machine regardless of the computer architecture. It is a robust, secure, interpreted, threaded and a dynamic language. The idea behind developing Java was "Write once, run anywhere". Java also has the maximum number of natural language processing toolkit implementations as compared to any other language. This is a good option to have for exploratory purposes. Some of the other popular natural language processing toolkits in Java are Mallet [79], OpenNLP and Stanford NLP.

For graphics and plotting we used the ggplot2 graphics package [80] for R stats. R stats is a free programming language and software environment for statistical computing and graphics. It is a very popular language among data miners. It is a very easy to use scripting language and nicely implemented statistics libraries at the core.

The goal of the study has been to conceptualize and demonstrate how manual curation of protein-protein interactions can be efficiently speeded up. And thus, we chose a decently performing classifier for evaluation. The aim has not been to run the same data against multiple different classifiers to identify the best performing classifier. We wanted to demonstrate that a highly manual task of protein-protein interaction curation could be performed faster with use of text mining. Due to the noise in textual data, along with the complexity of use of natural language processing techniques the expert data curators have not been able of seriously consider the automated approach. Making it easy to apprehend and demonstrating the potential of such an automated approach has been our aim.

CHAPTER SIX: CONCLUSION

In this study we have presented a context specific mining approach for annotating protein interactions with experimental evidence. Methods used so far for extracting protein interaction information have reported binary interactions between pairs of proteins. With this approach we support these binary interactions with experimental evidence. The performance of any machine learning system depends on the availability of data. The more the data, the better the system would perform. Though, our system does not have a very high performance, we have demonstrated that such an approach can help in the task of manual curation. In the future, we want to develop efficient scoring techniques to add confidence to the extracted information. Furthermore, we also want to annotate the interacting protein pair with more experimental information that just the detection methods. We want to build an incremental learning system that can adapt and learn continuously. Such an approach can continuously improve a system with high data bias by improving without having to train with the whole data every time.

REFERENCES

[1] A. Chatr-Aryamontri, A. Winter, L. Perfetto, L. Briganti, L. Licata, M. Iannuccelli, L. Castagnoli, G. Cesareni, and M. Tyers, "Benchmarking of the 2010 BioCreative Challenge III text-mining competition by the BioGRID and MINT interaction databases.," *BMC bioinformatics*, vol. 12 Suppl 8, no. Suppl 8, p. S8, Jan. 2011.

[2] R. Hashizume, M. Fukuda, I. Maeda, H. Nishikawa, D. Oyake, Y. Yabuki, H. Ogata, and T. Ohta, "The RING heterodimer BRCA1-BARD1 is a ubiquitin ligase inactivated by a breast cancer-derived mutation.," *The Journal of biological chemistry*, vol. 276, no. 18, pp. 14537–40, May 2001.

[3] A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. O'Donnell, T. Reguly, A. Breitkreutz, A. Sellam, D. Chen, C. Chang, J. Rust, M. Livstone, R. Oughtred, K. Dolinski, and M. Tyers, "The BioGRID interaction database: 2013 update.," *Nucleic acids research*, vol. 41, no. Database issue, pp. D816–23, Jan. 2013.

[4] A. Chatr-aryamontri, A. Ceol, L. M. Palazzi, G. Nardelli, M. V. Schneider, L. Castagnoli, and G. Cesareni, "MINT: the Molecular INTeraction database.," *Nucleic acids research*, vol. 35, no. Database issue, pp. D572–4, Jan. 2007.

[5] S. Kerrien, B. Aranda, L. Breuza, A. Bridge, F. Broackes-Carter, C. Chen, M. Duesbury, M. Dumousseau, M. Feuermann, U. Hinz, C. Jandrasits, R. C. Jimenez, J. Khadake, U. Mahadevan, P. Masson, I. Pedruzzi, E. Pfeiffenberger, P. Porras, A. Raghunath, B. Roechert, S. Orchard, and H. Hermjakob, "The IntAct molecular interaction database in 2012.," *Nucleic acids research*, vol. 40, no. Database issue, pp. D841–6, Jan. 2012.

[6] T. S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, L. Balakrishnan, A. Marimuthu, S. Banerjee, D. S. Somanathan, A. Sebastian, S. Rani, S. Ray, C. J. Harrys Kishore, S. Kanth, M. Ahmed, M. K. Kashyap, R. Mohmood, Y. L. Ramachandra, V. Krishna, B. A. Rahiman, S. Mohan, P. Ranganathan, S. Ramabadran, R. Chaerkady, and A. Pandey, "Human Protein Reference Database--2009 update.," *Nucleic acids research*, vol. 37, no. Database issue, pp. D767–72, Jan. 2009.

[7] A. Ceol, A. Chatr-Aryamontri, L. Licata, and G. Cesareni, "Linking entries in protein interaction database to structured text: The {FEBS} Letters experiment," *{FEBS} Letters*, vol. 582, no. 8, pp. 1171–1177, 2008.

[8] "Text Mining." [Online]. Available: http://en.wikipedia.org/wiki/Text_mining.

[9] S. Sohn, M. Torii, D. Li, K. Wagholikar, S. Wu, and H. Liu, “A hybrid approach to sentiment sentence classification in suicide notes.,” *Biomedical informatics insights*, vol. 5, no. Suppl. 1, pp. 43–50, Jan. 2012.

[10] M. Liakata, J.-H. Kim, S. Saha, J. Hastings, and D. Rebholz-Schuhmann, “Three hybrid classifiers for the detection of emotions in suicide notes.,” *Biomedical informatics insights*, vol. 5, no. Suppl. 1, pp. 175–84, Jan. 2012.

[11] J. P. Pestian, P. Matykiewicz, J. Grupp-Phelan, S. A. Lavanier, J. Combs, and R. Kowatch, “Using natural language processing to classify suicide notes.,” *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, no. June, p. 1091, Jan. 2008.

[12] J. Klekota, F. P. Roth, and S. L. Schreiber, “Query Chem: a Google-powered web search combining text and chemical structures.,” *Bioinformatics (Oxford, England)*, vol. 22, no. 13, pp. 1670–3, Jul. 2006.

[13] E. H. Shortliffe, “Computer programs to support clinical decision making.,” *JAMA : the journal of the American Medical Association*, vol. 258, no. 1, pp. 61–6, Jul. 1987.

[14] D. L. Hunt, R. B. Haynes, S. E. Hanna, and K. Smith, “Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review.,” *JAMA : the journal of the American Medical Association*, vol. 280, no. 15, pp. 1339–46, Oct. 1998.

[15] D. Demner-fushman, W. W. Chapman, and C. J. Mcdonald, “What can Natural Language Processing do for Clinical Decision,” vol. 42, no. 5, pp. 760–772, 2010.

[16] I. Androutsopoulos, J. Koutsias, K. V Chandrinos, G. Paliouras, and C. D. Spyropoulos, “An Evaluation of Naive Bayesian Anti-Spam Filtering,” pp. 9–17, 2000.

[17] T. Stone, “Parameterization of Na ¨ ıve Bayes for Spam Filtering,” 2003.

[18] V. P. Deshpande, R. F. Erbacher, and C. Harris, “An Evaluation of Naïve Bayesian Anti-Spam Filtering Techniques,” no. June, 2007.

[19] O. documentation, “Introduction to Support Vector Machines.”

[20] C. C. Aggarwal and C. Zhai, “Chapter 6 A SURVEY OF TEXT CLASSIFICATION ALGORITHMS.”

[21] L. Tari, S. Anwar, S. Liang, J. Cai, and C. Baral, “Discovering drug-drug interactions: a text-mining and reasoning approach based on properties of drug

metabolism.," *Bioinformatics (Oxford, England)*, vol. 26, no. 18, pp. i547–53, Sep. 2010.

[22] A. Gottlieb, G. Y. Stein, Y. Oron, E. Ruppin, and R. Sharan, "INDI: a computational framework for inferring drug interactions and their associated recommendations.," *Molecular systems biology*, vol. 8, no. 592, p. 592, Jan. 2012.

[23] M. Kuhn, C. von Mering, M. Campillos, L. J. Jensen, and P. Bork, "STITCH: interaction networks of chemicals and proteins.," *Nucleic acids research*, vol. 36, no. Database issue, pp. D684–8, Jan. 2008.

[24] D. M. Jessop, S. E. Adams, E. L. Willighagen, L. Hawizy, and P. Murray-Rust, "OSCAR4: a flexible architecture for chemical text-mining.," *Journal of cheminformatics*, vol. 3, no. 1, p. 41, Oct. 2011.

[25] L. Hawizy, D. M. Jessop, N. Adams, and P. Murray-Rust, "ChemicalTagger: A tool for semantic text-mining in chemistry.," *Journal of cheminformatics*, vol. 3, no. 1, p. 17, Jan. 2011.

[26] B. Kolluru, L. Hawizy, P. Murray-Rust, J. Tsujii, and S. Ananiadou, "Using workflows to explore and optimise named entity recognition for chemistry.," *PloS one*, vol. 6, no. 5, p. e20181, Jan. 2011.

[27] J. a Townsend, S. E. Adams, C. a Waudby, V. K. de Souza, J. M. Goodman, and P. Murray-Rust, "Chemical documents: machine understanding and automated information extraction.," *Organic & biomolecular chemistry*, vol. 2, no. 22, pp. 3294–300, Nov. 2004.

[28] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, "PubChem: a public information system for analyzing bioactivities of small molecules.," *Nucleic acids research*, vol. 37, no. Web Server issue, pp. W623–33, Jul. 2009.

[29] T. L. Griffiths and M. Steyvers, "Finding scientific topics.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl , no. Suppl 1, pp. 5228–5235, 2004.

[30] H. Wang, Y. Ding, J. Tang, X. Dong, B. He, J. Qiu, and D. J. Wild, "Finding Complex Biological Relationships in Recent PubMed Articles Using Bio-LDA," *PLoS ONE*, vol. 6, no. 3, p. 14, 2011.

[31] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The Database of Interacting Proteins: 2004 update.," *Nucleic acids research*, vol. 32, no. Database issue, pp. D449–51, Jan. 2004.

[32] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell, “Protein-protein interaction networks and biology--what’s the connection?,” *Nature biotechnology*, vol. 26, no. 1, pp. 69–72, Jan. 2008.

[33] S. Kerrien, S. Orchard, L. Montecchi-Palazzi, B. Aranda, A. F. Quinn, N. Vinod, G. D. Bader, I. Xenarios, J. Wojcik, D. Sherman, M. Tyers, J. J. Salama, S. Moore, A. Ceol, A. Chatr-Aryamontri, M. Oesterheld, V. Stümpflen, L. Salwinski, J. Nerothin, E. Cerami, M. E. Cusick, M. Vidal, M. Gilson, J. Armstrong, P. Woollard, C. Hogue, D. Eisenberg, G. Cesareni, R. Apweiler, and H. Hermjakob, “Broadening the horizon--level 2.5 of the HUPO-PSI format for molecular interactions.,” *BMC biology*, vol. 5, p. 44, Jan. 2007.

[34] L. M. Brettner and J. Masel, “Protein stickiness, rather than number of functional protein-protein interactions, predicts expression noise and plasticity in yeast.,” *BMC systems biology*, vol. 6, no. 1, p. 128, Jan. 2012.

[35] D. M. Eisenmann, “Wnt signaling (June 25, 2005), WormBook, ed. The C. elegans Research Community, WormBook, doi/10.1895/wormbook.1.7.1.” 2005.

[36] P. P. B. P. Thermo Scientific, “Pull-Down Assays.” .

[37] A. Cuadrado, V. Lafarga, P. C. F. Cheung, I. Dolado, S. Llanos, P. Cohen, and A. R. Nebreda, “A new p38 MAP kinase-regulated transcriptional coactivator that stimulates p53-dependent apoptosis.,” *The EMBO journal*, vol. 26, no. 8, pp. 2115–26, Apr. 2007.

[38] D. Rebholz-Schuhmann, A. Oellrich, and R. Hoehndorf, “Text-mining solutions for biomedical research: enabling integrative biology.,” *Nature reviews. Genetics*, vol. 13, no. 12, pp. 829–39, Dec. 2012.

[39] D. G. Jamieson, M. Gerner, F. Sarafraz, G. Nenadic, and D. L. Robertson, “Towards semi-automated curation: using text mining to recreate the HIV-1, human protein interaction database.,” *Database : the journal of biological databases and curation*, vol. 2012, p. bas023, Jan. 2012.

[40] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, “Automated extraction of information on protein–protein interactions from the biological literature,” *Bioinformatics*, vol. 17, no. 2, pp. 155–161, 2001.

[41] M. Huang, X. Zhu, Y. Hao, D. G. Payan, K. Qu, and M. Li, “Discovering patterns to extract protein–protein interactions from full texts,” *Bioinformatics*, vol. 20, no. 18, pp. 3604–3612, 2004.

[42] M. Craven and J. Kumlien, “Constructing biological knowledge bases by extracting information from text sources.,” *Proceedings / ... International*

Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology, pp. 77–86, Jan. 1999.

[43] J. Pustejovsky, J. Castafio, J. Zhang, M. Kotecki, and B. Cochran, "Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations," pp. 362–373.

[44] a Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii, "Event extraction from biomedical papers using a full parser.," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, vol. 419, pp. 408–19, Jan. 2001.

[45] G. Leroy, H. Chen, and J. D. Martinez, "A shallow parser based on closed-class words to capture relations in biomedical text," *Journal of Biomedical Informatics*, vol. 36, no. 3, pp. 145–158, 2003.

[46] J. C. Park, H. S. Kim, and J. J. Kim, "Bidirectional Incremental Parsing For Automatic Pathway Identification With Combinatory Categorial Grammar," pp. 396–407.

[47] D. Zhou, Y. He, and C. K. Kwoh, "Validating Text Mining Results on Protein-Protein Interactions Using Gene Expression Profiles."

[48] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski, "A graph kernel for protein-protein interaction extraction," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, 2008, no. June, pp. 1–9.

[49] M. He, Y. Wang, and W. Li, "PPI finder: a mining tool for human protein-protein interactions.," *PloS one*, vol. 4, no. 2, p. e4554, Jan. 2009.

[50] S. Agarwal, F. Liu, and H. Yu, "Simple and efficient machine learning frameworks for identifying protein-protein interaction relevant articles and experimental methods used to study the interactions.," *BMC bioinformatics*, vol. 12 Suppl 8, no. Suppl 8, p. S10, Jan. 2011.

[51] W. E. Winkler and U. S. Bureau, "The State of Record Linkage and Current Research Problems," 1962.

[52] X. Wang, R. Rak, A. Restificar, C. Nobata, C. J. Rupp, R. T. B. Batista-Navarro, R. Nawaz, and S. Ananiadou, "Detecting experimental techniques and selecting relevant documents for protein-protein interactions from biomedical literature.," *BMC bioinformatics*, vol. 12 Suppl 8, no. Suppl 8, p. S11, Jan. 2011.

[53] A. Lourenço, M. Conover, A. Wong, A. Nematzadeh, F. Pan, H. Shatkay, and L. M. Rocha, "A linear classifier based on entity recognition tools and a statistical

approach to method extraction in the protein-protein interaction literature.," *BMC bioinformatics*, vol. 12 Suppl 8, no. Suppl 8, p. S12, Jan. 2011.

[54] "A General Introduction to the E-utilities." National Center for Biotechnology Information (US), 26-May-2009.

[55] C. Kohlschütter, "boilerpipe." 2009.

[56] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, p. 441, 2010.

[57] L. Smith, T. Rindflesch, and W. J. Wilbur, "MedPost: a part-of-speech tagger for bioMedical text.," *Bioinformatics (Oxford, England)*, vol. 20, no. 14, pp. 2320–1, Sep. 2004.

[58] B. Carpenter and B. Baldwin, "LingPipe 4.1.0." 2008.

[59] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O'Reilly Media, 2009.

[60] G. D. Forney Jr, "The viterbi algorithm: A personal history," *arXiv preprint cs/0504020*, 2005.

[61] A. L. E. X. Y. Eh, J. A. H. Itzeman, and L. Y. H. Irschman, "Rapidly Retargetable Approaches to De-identification in Medical Records," 2007.

[62] a Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. a Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, a Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron, "Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.," *Nature genetics*, vol. 29, no. 4, pp. 365–71, Dec. 2001.

[63] M. Torii, Z. Hu, C. H. Wu, and H. Liu, "BioTagger-GM: a gene/protein name recognition system.," *Journal of the American Medical Informatics Association : JAMIA*, vol. 16, no. 2, pp. 247–55, 2009.

[64] H. Yu and E. Agichtein, "Extracting synonymous gene and protein terms from biological literature," *Bioinformatics*, vol. 19, no. Suppl 1, pp. i340–i349, Jul. 2003.

[65] A. Koike and T. Takagi, "Gene / protein / family name recognition in biomedical literature," vol. 1, pp. 9–16, 2004.

[66] L. Tanabe and W. J. Wilbur, “Tagging Gene and Protein Names in Full Text Articles,” no. July, pp. 9–13, 2002.

[67] D. Park, R. Singh, M. Baym, C.-S. Liao, and B. Berger, “IsoBase: a database of functionally related proteins across PPI networks,” *Nucleic Acids Research*, vol. 39, no. Database issue, pp. D295–D300, 2011.

[68] L. Tanabe, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur, “GENETAG: a tagged corpus for gene/protein named entity recognition.,” *BMC bioinformatics*, vol. 6 Suppl 1, no. Suppl 1, p. S3, Jan. 2005.

[69] P. Komarek and A. Calvet, “Logistic Regression for Data Mining and High-Dimensional Classification Committee Alan Frieze , Chair.”

[70] B. Rd, “More accurate tests for the statistical significance of result,” 2000.

[71] “Information Systems and Evaluations.” .

[72] “Precision and Recall,” *Wikipedia*. .

[73] “Measuring Search Effectiveness.”

[74] “Accuracy and precision.” Wikipedia.

[75] S. Lindsay, “Statistical Generalization.” SAGE Publications, Inc., pp. 893–894, 2010.

[76] E. Hatcher, O. Gospodnetic, and M. McCandless, “Lucene in action.” Manning Publications, 2004.

[77] D. Raber, *The problem of information: An introduction to information science*. Scarecrow Pr, 2003.

[78] B. Carpenter, “LingPipe for 99.99% recall of gene mentions,” in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 2007, vol. 23, pp. 307–309.

[79] A. K. McCallum, “MALLET: A Machine Learning for Language Toolkit.” 2002.

[80] H. Wickham, *ggplot2: elegant graphics for data analysis*. Springer Publishing Company, Incorporated, 2009.

APPENDICES

Appendix A: Ontology for interaction detection method with ID Term mapping

MI:0004	affinity chromatography technology
MI:0006	anti bait coimmunoprecipitation
MI:0007	anti tag coimmunoprecipitation
MI:0008	array technology
MI:0010	beta galactosidase complementation
MI:0011	beta lactamase complementation
MI:0012	bioluminescence resonance energy transfer
MI:0014	adenylate cyclase complementation
MI:0016	circular dichroism
MI:0017	classical fluorescence spectroscopy
MI:0018	two hybrid
MI:0019	coimmunoprecipitation
MI:0020	transmission electron microscopy
MI:0027	cosedimentation
MI:0028	cosedimentation in solution
MI:0029	cosedimentation through density gradient
MI:0030	cross-linking study
MI:0031	protein cross-linking with a bifunctional reagent
MI:0038	dynamic light scattering
MI:0040	electron microscopy
MI:0042	electron paramagnetic resonance
MI:0045	experimental interaction detection
MI:0046	experimental knowledge based
MI:0047	far western blotting
MI:0048	filamentous phage display
MI:0049	filter binding
MI:0051	fluorescence technology
MI:0052	fluorescence correlation spectroscopy
MI:0053	fluorescence polarization spectroscopy
MI:0054	fluorescence-activated cell sorting
MI:0055	fluorescent resonance energy transfer
MI:0065	isothermal titration calorimetry
MI:0066	lambda phage display
MI:0067	light scattering
MI:0069	mass spectrometry studies of complexes
MI:0071	molecular sieving

MI:0077	nuclear magnetic resonance
MI:0081	peptide array
MI:0084	phage display
MI:0089	protein array
MI:0091	chromatography technology
MI:0096	pull down
MI:0097	reverse ras recruitment system
MI:0099	scintillation proximity assay
MI:0104	static light scattering
MI:0107	surface plasmon resonance
MI:0108	t7 phage display
MI:0111	dihydrofolate reductase reconstruction
MI:0112	ubiquitin reconstruction
MI:0114	x-ray crystallography
MI:0115	yeast display
MI:0226	ion exchange chromatography
MI:0227	reverse phase chromatography
MI:0228	cytoplasmic complementation assay
MI:0229	green fluorescence protein complementation assay
MI:0230	membrane bound complementation assay
MI:0231	mammalian protein protein interaction trap
MI:0254	genetic interference
MI:0276	blue native page
MI:0369	lex-a dimerization assay
MI:0370	tox-r dimerization assay
MI:0397	two hybrid array
MI:0398	two hybrid pooling approach
MI:0399	two hybrid fragment pooling approach
MI:0402	chromatin immunoprecipitation assay
MI:0404	comigration in non denaturing gel electrophoresis
MI:0405	competition binding
MI:0406	deacetylase assay
MI:0410	electron tomography
MI:0411	enzyme linked immunosorbent assay
MI:0412	electrophoretic mobility supershift assay
MI:0413	electrophoretic mobility shift assay
MI:0415	enzymatic study
MI:0416	fluorescence microscopy
MI:0417	footprinting
MI:0419	gtpase assay
MI:0420	kinase homogeneous time resolved fluorescence

MI:0423	in-gel kinase assay
MI:0424	protein kinase assay
MI:0426	light microscopy
MI:0428	imaging technique
MI:0432	one hybrid
MI:0434	phosphatase assay
MI:0435	protease assay
MI:0437	protein tri hybrid
MI:0440	saturation binding
MI:0510	homogeneous time resolved fluorescence
MI:0515	methyltransferase assay
MI:0516	methyltransferase radiometric assay
MI:0588	3 hybrid method
MI:0605	enzymatic footprinting
MI:0655	lambda repressor two hybrid
MI:0657	systematic evolution of ligands by exponential enrichment
MI:0663	confocal microscopy
MI:0676	tandem affinity purification
MI:0678	antibody array
MI:0697	dna directed dna polymerase assay
MI:0700	rna directed rna polymerase assay
MI:0726	reverse two hybrid
MI:0728	gal4 vp16 complementation
MI:0729	luminescence based mammalian interactome mapping
MI:0807	comigration in gel electrophoresis
MI:0808	comigration in sds page
MI:0809	bimolecular fluorescence complementation
MI:0825	x-ray fiber diffraction
MI:0826	x ray scattering
MI:0841	phosphotransfer assay
MI:0858	immunodepleted coimmunoprecipitation
MI:0859	intermolecular force
MI:0870	demethylase assay
MI:0872	atomic force microscopy
MI:0880	atpase assay
MI:0889	acetylation assay
MI:0892	solid phase assay
MI:0920	ribonuclease assay

Appendix B: Running the script

The data and the code is put in `/home/bibliomix/ypandit/PpiAnnotator` on `regen.informatics.iupui.edu`

About the tool

- There are 2 versions of the tool
    1. One that runs with full-text
    2. One that runs with extracted features (Recommended. See Notes at the end).

1. With full-text data
    - The system extracts features on the fly, and all the features are held up in the memory.
    - *Advantage:* Once the model is saved after training, it can be used to classify any text document
    - *Disadvantage:*
        1. Memory usage is very high, due to in-memory loading of features.
        2. The feature chunks are extracted dynamically as the training runs. Hence, the classifier object is not `Serializable`. So it cannot be saved to a file.

2. With features

- The features from each training and testing document are extracted and saved to individual files (identified by PMID).
- *Advantage:* Works well with controlled set of training and testing data. Should run on low memory system.
- *Disadvantage:* However, if a text document is to be classified, the features of that particular document are required to be in the cache that is used for training and testing. This is because; the data is identified by the PMID and not by the text itself.

For this study, we have used the approach with features as the training and testing data are a controlled set

Configuration

Files provided on the server:

1. PpiExtraction.jar
2. . config.properties
3. data/
    - full/
        1. train/
        2. test/
    - features/
        1. train/
        2. test/
    - search/
    - model/
        1. ne-en-bio-genia.TokenShapeChunker
        2. bio-exp_factors.CharLmHmmChunker
        3. ne-efo.Dictionary
        4. mi_imt_map.txt
4. categories.txt

Instructions are taken from config.properties. This file has 9 input parameters

1. data_type - This can be either full or features
    - full will expect train and test data as full-text articles
    - features will expect train and test data in the form of extracted features
2. iterations - Integer value for number of epochs in train/test
3. cross_folds - Integer value for number of folds for validation in train/test
4. verbose - Boolean value to set the train/test process in verbose mode or not
5. train_data - Path to train data folder
6. test_data - Path to test data folder
7. categories_file - Path to file with categories to train and test
    - Having this file helps in training/testing different classes without having to modify the train/test dataset
    - Only the data for classes/categories mentioned in the file with be picked up from train_data and test_data
8. model - File name/path to save the trained model. Model cannot be written to file when using the data_type=full
9. action - This can be either train or annotate
    1. train - This action will train and test the system against the given dataset
        - All the above parameters are required for this action to run, or else the system will break throwing some error.
    2. annotate - This action will use the trained model to predict top few classes for given text (by PMID). Using the predicted class, identified proteins

and part-of-speech composition of the sentence, it will search for sentence based evidences from the text

- Only test_data and model are required parameters to run this action

---

How to

For a given $config.properties$ which looks like this

$$
\begin{aligned}
&\#\ Type\ of\ data\\
&data\_type = features\\
\\
&\#\ train, annotate\\
&action = train\\
\\
&\#\ Train\ /\ test\ data\\
&train\_data = data/features/train\\
&test\_data = data/features/test\\
\\
&\#\ Categories\ file\\
&categories\_file = categories.txt\\
\\
&\#\ Train\ parameters\\
&iterations = 100\\
&cross\_folds = 2\\
&verbose = false\\
\\
&\#\ File\ to\ save\ the\ model\ to\\
&model = bio - imt.LRClassifier
\end{aligned}
$$

the command to run

$$java\ -jar\ PpiExtraction.jar\ config.properties$$

With the above command and config, it will print something like the following on the terminal

$\#\ categories$: 26
$\#\ folds$: 2
$Train\ data\ size$: 4315
$Feature\ cache\ size$: 2114
$Cache\ size$: 2114
$FOLD = 0 \quad ACC = 0.9372704254484505 \ \ +/-0.01637024991072757$
$FOLD = 1 \quad ACC = 0.9369430384258929 \ \ +/-0.01621875927745846$
$FOLD = 2 \quad ACC = 0.9369430384258929 \ \ +/-0.01621875927745846$

And on a modified config, to run $action = annotate$ on a single full-text document of a

set of full-text documents

$\#\ Type\ of\ data$
$data\_type = features$

$\#\ train, annotate$
$action = annotate$

$\#\ Train\ /\ test\ data$
$\#\ train\_data =$
$test\_data = data/search/$

$\#\ Categories\ file\ \#$
$categories\_file =$

$\#\ Train\ parameters$
$\#\ iterations =$
$\#\ cross\_folds =$
$\#\ verbose =$

$\#\ File\ to\ save\ the\ model\ to$
$model = bio - imt.LRClassifier$

the output on running $java\ -jar\ PpiExtraction.jar\ config.properties$ should look

something like

$Searching\ for\ PPI\ evidences\ in\ 18337465.txt$

$Sentence\ ID$: 15

$Sentence$: $Yeast\ two-hybrid\ screening\ using\ a\ portion\ of\ UNC$

$-89\ including\ PK2, yielded\ SCPL-1\ (small\ CTD\ phosphatase-like$

$-1), which\ contains\ a\ C\ terminal\ domain\ (CTD)\ phosphatase\ type\ domain.$

$Score$: 1.5956881

IMT: $two\ hybrid\ Proteins$: $PK2, SCPL-1, small\ CTD\ phosphatase-like$

$-1, C\ terminal\ domain, CTD)\ phosphatase\ type\ domain,$

$Sentence\ ID$: 217

$Sentence$: $To\ determine\ which\ portions\ of\ the\ UNC$

$-89\ bait\ are\ minimally\ required\ to\ interact\ with\ SCPL$

$-1, deletion\ derivatives\ of\ the\ segment\ Ig-Fn3$

$-PK2\ were\ tested\ by\ two-hybrid\ against\ SCPL-1a\ and$

$-1b\ full-length\ prey.$

$Score$: 1.5956881

IMT: $two\ hybrid$

$Proteins$: $SCPL-1, SCPL-1a\ and\ -1b, prey,$

$Searching\ for\ PPI\ evidences\ in$ 19088068.txt

$Sentence\ ID$: 254

$Sentence$: $SDS-PAGE\ (14\%\ gel)\ and\ Western\ blotting\ analysis\ of\ H2AZ\ pull$

$-down\ by\ SWR1(1-\ 681)\ or\ SWR1(\square N2)\ complexes\ at\ the\ 0.2\ or\ 0.3\ M\ KCl\ condition.$

$Score$: 2.1836114

$IMT: pull\ down$

$Proteins: SWR1, SWR1, N2)\ complexes,$

$Sentence\ ID: 334$

$Sentence: Recent\ studies\ have\ shown\ that\ an\ N$

$-\ terminal\ subdomain\ (residues\ 340 -\ 411) of\ Swr1, the\ HSA\ domain, is\ sufficient\ to\ pull$

$-\ down\ Arp4\ and\ Act1\ (42), and\ thus\ can\ be\ considered\ as\ a\ binding\ platform\ for$

$Arp4\ and\ actin.$

$Score: 2.1836114$

$IMT: pull\ down$

$Proteins: N$

$-\ terminal\ subdomain\ (residues\ 340 -\ 411, Swr1, HSA\ domain, Act1, Arp4,$

Notes:

1. Data is imbalanced, so
    - Different combinations of cross_folds and iterations will show variations in performance.
    - For better evaluation, many different combinations of cross_folds & iterations along with varied number of categories in the categories_file have been run.

- For some of the runs, the data in categories/classes with low data was extrapolated. And the data for categories with very high distribution was reduced.
- Final performance reported is an micro-average of many such different runs.

2. The training and testing using full-text is very computationally exhaustive as the features are extracted and held in memory for analysis.
3. To assign more memory to java, add -Xmx5000m (max memory of 5GB) to the command line arguments.