PREDICTION BY PARTIAL MATCHING FOR IDENTIFICATION OF BIOLOGICAL ENTITIES

Arvind Kumar Thirumalaiswamy Sekhar

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 2008

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**

Malika Mahoui, Ph.D., Chair

Narayanan Perumal, Ph.D.

Pedro Romero, Ph.D.

The work presented here is dedicated to my mother and sister whose support has brought me this far. I would also like to dedicate this to all my teachers and friends who have been reasons for my scientific curiosity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I express my sincere thanks to Dr. Malika Mahoui who has been my guide and mentor at Indiana University and who has been the source of inspiration for this thesis. I also express my heartfelt gratitude to Dr. Narayanan Perumal and Dr. Pedro Romero for being a part of my thesis committee and for having provided such tremendous support throughout the course of my study. I extend my appreciation and thanks to Dr. William Teahan, University of Wales at Bangor for providing valuable suggestions and constantly motivating me for improvements on the methods used in this thesis. I wish to acknowledge the support of the faculty and staff of the Indiana University School of Informatics, Indianapolis. I express honest thanks for all who have supported me through these years of study. The inconsistencies remain my own.

ABSTRACT

Arvind Kumar Thirumalaiswamy Sekhar

PREDICTION BY PARTIAL MATCHING FOR IDENTIFICATION OF BIOLOGICAL ENTITIES

As biomedical research and advances in biotechnology generate expansive datasets, the need to process this data into information has grown simultaneously. Specifically, recognizing and extracting these “key” phrases comprising the named entities from this information databank promises a plethora of applications for scientists. The ability to construct interaction maps, identify proteins as drug targets are two important applications. Since we have the choice of defining what is “useful”, we can potentially utilize text mining for our purpose. In a novel attempt to beat the challenge, we have put information theory and text compression through this task. Prediction by partial matching is an adaptive text encoding scheme that blends together a set of finite context Markov models to predict the probability of the next token in a given symbol stream. We observe, named entities such as gene names, protein names, gene functions, protein-protein interactions – all follow symbol statistics uniquely different from normal scientific text. By using well defined training sets that allow us to selectively differentiate between named entities and the rest of the symbols; we were able to extract them with a good accuracy. We have implemented our tests, using the Text Mining Toolkit, on identification of gene functions and protein-protein interactions with f-scores (based on precision & recall) of 0.9737 and 0.6865 respectively. With our results, we foresee the application of such an approach in automated information retrieval in the realm of biology.

1. INTRODUCTION

The use of high throughput techniques for characterizing biological entities has attained rapid progress in recent years and even lead to the subsequent accumulation of scientific literature (Total number of PubMed abstracts is 9437082 as on January 6, 2008) (NCBI). Eventually researchers have turned their attention towards information retrieval, information extraction and text mining over these growing literature databases. Benefitting from this intense computational information extraction research focus are biologists, bioinformaticians, database curators and even medical scientists to a certain extent (Blaschke et al, 2002).

A quick look at the number of searches performed each year on these scientific articles just with PubMed (Figure 1) clearly indicates to us the growing dependence of biologists, among other users of PubMed, on these literature warehouses for obtaining quality information. It would not be surprising therefore that there is a need to develop a very efficient system that can automatically annotate the information within these articles providing for a more accurate information retrieval. However, one has to note that the annotation would naturally stem from the ability to extract named entities, depending on whatever we define them to be, from these articles with a certain degree of accuracy.

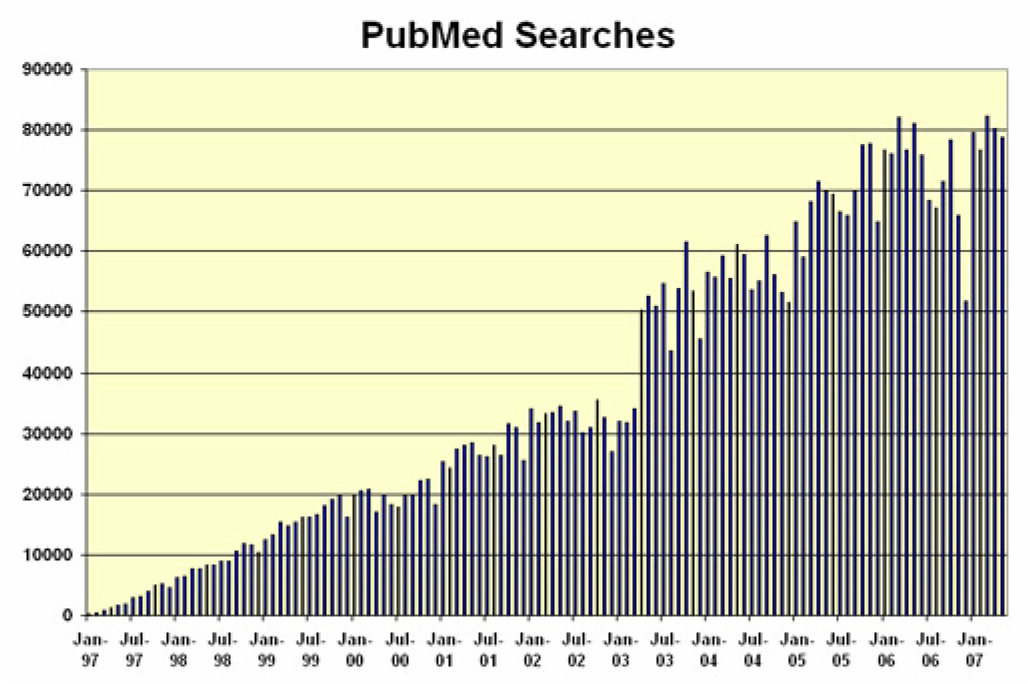


Figure 1: Number of PubMed searches executed between 1997 and 2007(NCBI).

While many attempts have been made so far in achieving this through natural language processing (Fukuda et al 1998, Collier et al, 2000, Tanabe et al, 2002), dictionary based approach (Krauthammer et al, 2004), rule based approach (Krallinger et al, 2005, Seki and Mostafa, 2003), and even machine learning approaches using Support Vector Machines (Lee et al, 2003) , we take a novel approach to this challenge and attempt to return a solution using a data compression technique. Text compression has been widely studied and the use of adaptive statistical encoding techniques for text compression and their efficiency has already been well proven (Cleary and Witten, 1984). In particular, our inspiration to use this method comes from the observation that Prediction by Partial Matching (PPM) has been successfully applied in token characterization as in characterization of text to belong to one language or the other (Celikel, 2005). PPM uses finite context Hidden Markov Model and blends together some of these models to achieve the prediction of the probability with which a specific symbol would appear at a given position in an input stream. The advantage is that these models exploit the statistics associated with the specific symbol set that form the named entities. Therefore a careful selection of the training sets can lead to a very good discrimination of a token as belonging to one class of symbols.

In this thesis we work to compose suitable training sets to achieve supreme performance measures in extracting key phrases from biomedical text. Our intent was to examine the success of using a text compression based text mining approach and study the same over two varied entities and observe the merits and demerits from them. The thesis itself is outlined to indicate the various terms and concepts that appear in this study, the methods involved in composing datasets for training and testing for the two tasks of identifying gene function phrases and protein-protein interaction phrases, the key differences in the results observed and the causes and solutions to some of the factors affecting the performance of the system.

We have used PPM to identify named entities and key phrases including gene functions and protein -protein interactions to study the suitability of this approach to tackling the problem of text mining in biology. Our results indicate that a very clear and specific definition of the named entity may be essential for the complete success of this approach. The technique seems very promising and if used with appropriate training sets for problems that do not require semantic interpolations it will prove to be a good success.

More importantly, the advantage of this method can be well comprehended by looking at the numerous applications and benefits that can be derived by using such a technique over literature data. Some of the key benefits can be enlisted as:

- Efficient information retrieval
- Advanced database curation
- Ease of annotation
- Save time and effort
- Improved data mining and new relation discovery
- Extension to useful entities such as Protein-Protein interactions

These benefits have resounding implications in the field of biology and medicine. One may think , intuitively, that the ability to identify protein-protein interactions will lead to improved construction of protein interaction networks and therefore a better understanding of how drugs may affect the network as a whole. Not to forget the value of identifying key terms themselves, such as drug names/protein names etc reducing the time involved in valuable research associated with these entities. Moreover, the ability to identify new relations is very significant as human errors in manual curation of these articles could lead to non identification of these otherwise useful entities.

2. BACKGROUND

2.1 Evaluation of Present Systems

Existing systems for text mining in the biomedical domain attempt to extract protein names/gene names or even protein – protein interactions from scientific text. While a majority of these approaches combine the extracted terms with the second phase of assigning Gene Ontology identifiers, we look to solving the extraction of named entities as is. With the task of GO based gene function identification established by the (BioCreative) competition, these approaches determine GO terms as a mandatory subtask. The methods that have been established fall under techniques making use of a dictionary based approach (Krauthammer et al, 2004), a machine learning approach (Stoica and Hearst, 2006, Ehler and Ruch, 2004) or other rule based approaches (Krallinger et al, 2005, Seki and Mostafa, 2003). The application of Natural Language Processing to gene/protein identification from biomedical text has been described (Fukuda et al 1998, Collier et al, 2000, Tanabe et al, 2002). (Raychaudhari et al) have shown that abstracts can be classified into major GO ID classes using maximum entropy, Naïve Bayes or nearest neighbor method. There have been attempts at identifying gene function relationships based on syntactic dependency (Kim and Park, 2004) and using sentence similarity through Naïve Bayes (Chiang and Yu, 2003). (Koike et al) have utilized shallow parsing and sentence structure analysis to perform automatic extraction of gene biological functions. They report a precision of 92% but a low recall of 60%. The Medical Knowledge Explorer (MeKE) system uses a Gene name Lexicon with a maximal coverage followed by a sentence alignment procedure, utilizing sentence structure analysis to identify functions of gene products(Chiang and Yu, 2003). Compared to these systems, our approach to gene function identification has been far more superior in performance as it generates a precision of 84.81% and a recall of 94.36%. Protein-protein interactions have not yet had the same kind of success rate as those of systems applied to gene function identification. (Ahmed et al) have developed an automated interaction identifier that performs sentence splitting followed by entity tagging to identify interaction phrases. They report a 26.94% precision and a 65.66% recall on the DIP dataset. Maximum entropy based approaches for part of speech tagging and their state of the art accuracy has been established (Ratnaparkhi et al, 1994). The use of synonym based entity recognition followed by a SVM based post processing applied to the BioCreative Subtask is described in (Fundel et al, 2005). The method focuses on just the identification of protein names and not the interactions. This could potentially be a part of a suitable post processing phase which could identify protein names. A synonym based identification coupled to entity tagging when applied to protein interaction identification resulted in a 47% recall and 93% precision (Otasek et al, 2006) .The precision and recall of our system which currently are 60.125% and 80% respectively provide a glimpse of the performance improvements that can be achieved.

2.2 Terms and Concepts

Text Mining

Witten, Bray, Mahoui and Teahan define text mining as "The process of analyzing text to extract information that is useful for particular purposes" (Witten et al, 1999). This definition of text mining is a very appropriate one for the problem we are trying to address. Indeed we have the choice of defining what is useful; we can extract selective entities from different text streams for a variety of purposes. Some examples of text mining would be finding words in Chinese text, finding names and addresses in bibliographic references, or classifying text by language, authorship, or genre.

Information Theory

Information theory is a branch of applied mathematics that deals with techniques to quantify information. The field came into existence formally after the work of Claude E Shannon was published in his paper "A Mathematical Theory of Communication" (Shannon, 1948). Information theory encompasses concepts from fields of science and engineering such as adaptive systems, anticipatory systems, artificial intelligence, complex systems, complexity science, cybernetics, informatics, machine learning. Concepts in information theory have parallels in human communication. Two essential features of "language" for communication form the pillars of information theory. First, it can be clearly observed that in any given language, words that are more frequently used are shorter than words that are less frequently used. The reason for is that sentences become much shorter to construct. For example, words such as "I", "the" appear more in common sentences than words such as "apple", "economics" etc. Second, any receiver should be able to identify the meaning of a sentence even if part of the sentence is distorted due to noise. This, one has to note, does not have any correlation to the "quality of the meaning" implied by the sentence. A sentence such as "How are you?" is as equivalent as a sentence such as "call 911" though the latter has a more significant import in the context.

Information theory seeks to provide insights into the classical problem of communicating information in the presence of noise and uncertainty. This usually involves communication over a noisy channel. When data is transmitted across a noisy channel, the ability to reconstruct the data with a certain degree of loss in fidelity is of at most importance. The actual uncertainty in the information during transmission came to be measured by a quantity known as entropy. Formally, entropy would be quantified in terms of the number of bits (or a suitable system) required to represent the given information. In general, greater the entropy, larger the bits and vice versa. Shannon showed, in his work, that, if the number of bits were to be lesser than the channel capacity then information could be transmitted successfully across the channel. This paved the way for the emergence of source coding and the other compression schemes outlined below.

Text Compression

Data and text compression refers to the use of encoding to represent the input stream using fewer bits (or any other suitable representation) resulting in lesser storage space and lesser transmission time (Lelewer and Hirschberg, 1987). One popular instance of compression with which many computer users are familiar is the ZIP file format, which, as well as providing compression, acts as an archive tool, storing many files in a single output file. Data compression techniques require that both the sender and the receiver involved in the communication be aware of the compression method being used. Though compression seems to solve storage and space problems, it really involves tradeoffs on the requirement of expensive equipment on the receiver end, say, and the actual overhead of transmitting the information as is. Also, most compression techniques work well when there is some kind of a statistical redundancy in the input stream. In the absence of any statistical redundancy, it would not be a surprise to note that the compression algorithm actually results in data expansion.

Lossless Data Compression

As the name suggests, lossless data compression works to compress the input stream in a manner that allows its reconstruction perfectly without any loss of information. Lossless compression schemes are the ones that mostly make use of statistical patterns in the data. These are applied more frequently to textual data such as word files, spreadsheets etc as the loss of information from these files is intolerable. Some examples of lossless compression schemes would be run length encoding, Lempel Ziv encoding (Lempel and Ziv, 1977), DEFLATE etc.

Lossy Data Compression

In contrast to Lossless Data Compression, this method allows a certain degree of loss in fidelity when the original data is reconstructed from the compressed source. This technique is useful in applications such as image and video compression where minor alterations remain almost imperceptible. In other words, Lossy Compression schemes, achieve a tradeoff between certain features whose alterations are very less perceived as opposed to the amount of compression achieved. For example, the human eye can tolerate changes in luminance more than changes in the color of images. Therefore certain compression algorithms, achieve a good compression ratio by accommodating a modification in the luminance values.

“Most of the data compression methods in common use today fall into one of two categories: dictionary-based schemes and statistical methods. In the world of small systems, dictionary based data compression techniques seem to be more popular at this time. However, by combining arithmetic coding with powerful modeling techniques, statistical methods for data compression can actually achieve better performance.” (Wikipedia).

Dictionary-Based Data Compression

Certain encoding schemes use a “fixed length” code to represent blocks of data in the input stream. These schemes whose encoding symbol set have the same length and are independent of the statistical frequency of input symbols are known as dictionary based methods. A very popular dictionary based encoding model is the LZW encoding model. As a working example, let us consider the ASCII code used to represent characters in the input text.

The ASCII symbol set has 256 characters to represent from the Standard English alphabet. Since a dictionary based approach uses the same length of code for each alphabet, we would need to fix a minimum length for this code that can be used to represent each of the 256 characters without ambiguity. Using the BIT representation of 1's and 0's we can see that each bit can accommodate 2 characters. Therefore the following table (Table 1) indicates how we can arrive at a code length for all the 256 characters to be encoded.

Code length (number of symbols)	**Number of characters accommodated**
1	2^1 =2
2	2^2 =4
3	2^3 =8
4	2^4 =16
5	2^5 =32
6	2^6 =64
7	2^7 =128
8	2^8 =256

Table 1: Number of Bits required for encoding

Thus we can observe that all symbols have equal code length and this may be arrived by using the minimum number of bits required to represent the entire alphabets without ambiguity.

Though simple to understand and easy to accomplish, the dictionary based methods work well for small systems only. The approach was rendered naïve by a look into Shannon's entropy theory which clearly suggests that by adopting a method which would vary the length of the code depending on the symbol frequency, we can greatly enhance the compression ratio.

Statistical Data Compression

Another, lossless compression scheme, the statistical data compression method works by encoding the input stream of symbols using a "variable length" code system contrary to the fixed length code approach of the dictionary based method (Rooij, 2003). The length of the code used to represent any symbol is directly related to the frequency or probability of occurrence of that symbol in the given text. Thus methods falling in this category use the statistical pattern in the data. It is intuitive to understand that by using the shorter codes for frequently occurring entities, we can achieve a better compression ratio.

Entropy

One definition of entropy equates it to the degree of disorder in a system (Sommers et al, 2004). Ordered and disordered states of a system are relative and a system in which randomness is maximal is called as a maximally disordered system. As we move further away from randomness to non random behavior, we move towards an ordered state. For example, a deck of cards which is shuffled and in which numbers appear in any random order is said to be maximally disordered. In the same manner a deck of cards which is arranged according to some pattern say the color of the card is said to be moving towards an ordered state. Thus by using entropy as a degree of disorder, we can measure how far from randomness a system actually is.

Information Entropy

Extending the concept of entropy to information theory, Shannon defined information entropy as the average message length (number of bits) required to communicate the true message in the given data is called the entropy. Intuitively, it is also the average information content that a receiver is missing when the true value of a random variable is not known.

The information entropy of a discrete random variable X, that can take on possible values $\{x_1 \ldots x_n\}$ is

$H(X) = -\sum p(x_i) \log_2 p(x_i)$

Where

$p(x_i) = \Pr(X=x_i)$ is the probability mass function of X.

Consider tossing a coin with known, not necessarily fair, probabilities of coming up heads or tails.

Since the result of the toss is unknown, the entropy of this event is maximal. This is equivalent to stating that the probability of an outcome is equally likely as another and hence there is no bias towards one result. This situation, as a consequence, requires a greater number of bits (1 in this case) to deliver the full information.

However, if we know the coin is not fair, and that the outcome is biased towards either heads or tails, then the uncertainty is reduced. The entropy that measures this uncertainty would be expected to lower down as the number of bits required to present the information is reduced.

The extreme case is that of a double-headed coin which never comes up tails. Then there is no uncertainty. The entropy is zero: each toss of the coin delivers no information.

Minimum Description Length

"The fundamental idea behind the MDL Principle is that any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally." (Grünwald, 1996).Since our hypothesis must capture the most regularity in the data; the hypothesis is the one that achieves maximum compression.

So far we can see that lossless statistical encoding works best in two general steps. First a statistical model for the data stream is generated and second, based on this model, encoding is performed. However the way in which the statistical model is used can further lead to two sets of encoding procedures. They are:

Non Adaptive Encoding

In a non adaptive encoding scheme, the statistic table is scanned only once and after this single pass, an encoding scheme is directly applied. This essentially means that the model statistic has to be also coupled with the actual encoding and transmitted to the decoder.

This will make sure that both the encoder and the decoder use the same model. Though this works fine in simple case, when large statistical models are generated to achieve better compression, the extra baggage of the large model to be transmitted can sideline any improvements in the compression itself. Non adaptive encoding is suitable only for small statistical models.

Adaptive encoding

In order to overcome the model addition problem of the non adaptive encoding procedure, the adaptive encoding technique was devised. The solution is that instead of making a single pass over the input stream to create a statistical model, one can allow both the encoder and decoder start with their statistical model in the same state. Each of them processes a single character at a time, and updates their models after the character is read in. This is very similar to the way most dictionary based schemes such as LZW coding work. A slight amount of efficiency is lost by beginning the message with a non-optimal model, but it is usually more than made up for by not having to pass any statistics with the message.

Huffman Encoding

Huffman encoding was the outcome of an assignment by Huffman in response to the challenge proposed by one of his professors at MIT. The algorithm generates a prefix free tree producing codes to represent symbols in the given text/data. Huffman algorithm is so prevalent in modern computer science and data compression that it is even used interchangeably with the encoding procedure itself (Huffman, 1952).

Huffman encoding can be easily understood by considering the following example.

Given a set of symbols and their weights (probabilities) for an input alphabet $A=\{a_1,a_2\ldots a_n\}$ with weights $W=\{w_1,w_2\ldots w_n\}$ we need to find codes $C=\{c_1,c_2\ldots c_n\}$ such that $L(c)=\sum w_i * \text{length}(c_i)$ must be the most optimal code length satisfying $L(c)<=L(t)$ for any $t(a, w)$.

Consider the following data table (Table 2)

Input	**Symbol**	**a**	**b**	**c**	**d**	**e**	**Sum**
	Weights	0.10	0.15	0.30	0.16	0.29	1
Output	**Codeword**	000	001	10	01	11	
	Code length	3	3	2	2	2	
	Path length	0.3	0.45	0.6	0.32	0.58	2.25
Optimality	**Probability budget**	0.125	0.125	0.25	0.25	0.25	1.00
	Information content(bits)	3.32	2.74	1.74	2.64	1.79	
	Entropy	0.332	0.411	0.521	0.423	0.518	2.205

Table 2: Example implementation of Information Entropy (adapted from wikipedia)

We are given five symbols a, b, c, d, e whose frequencies or probability of occurrence is 0.10, 0.15, 0.30, 0.16 and 0.29 respectively. Using the logic of assigning greater code length for a less frequent symbol we assign code words of lengths 3 to a, b and of lengths 2 to c, d and e. The average path length then turns out to be 2.25.

From the above table we can derive that Shannon's entropy for the given input symbols is 2.205. This is the entropy that is least possible for the given input set. Using Huffman coding we can see that we can generate an optimal code length of 2.25 which is very close to Shannon's entropy minimum.

A peculiar disadvantage of the Huffman encoding procedure is that, it always assigns an integer number of bits as a code length. Therefore if a symbol had a probability of say 90% then the $-\log_2 P$ is equal to 0.15 for the symbol. Huffman encoding can only assign either 0 or 1 bit to this symbol and assuming that it assigns 1 bit, we end up with a six times larger code. Therefore, Huffman coding had to make way for arithmetic encoding which would utilize floating point operations to generate a code to represent the symbol.

Arithmetic Encoding

Arithmetic encoding uses a procedure that assigns a single floating point number to a given symbol set there by encoding it in a non ambiguous and efficient manner devoid of the integer problem that Huffman encoding faces. Arithmetic encoding can be established by considering the example "Bill Gates" (Nelson, 1991). The distributions necessary to perform the actual encoding can be obtained as seen in (Table 3) (Table 4).

Character	**Frequency of occurrence**	**Probability**
b	1	1/10
i	1	1/10
l	2	2/10
g	1	1/10
a	1	1/10
t	1	1/10
e	1	1/10
s	1	1/10

Table 3: Probability distribution for input stream (adapted from wikipedia)

Each character is assigned a range of values between 0 and 1 as follows.

Character	**Frequency of occurrence**
b	0.10 -0.20
i	0.20-0.30
l	0.30-0.50
g	0.50-0.60
a	0.60-0.70
t	0.70-0.80
e	0.80-0.90
s	0.90-1.00

Table 4: Estimated Probabilities (adapted from wikipedia)

Space occupies the 0.0 to 0.10 range.

Our encoding would proceed as

Character	**Start value**	**End value**
B	0.10	0.20
I	0.12	0.13
L	0.123	0.125
L	0.1233	0.1235
space	0.12330	0.12331
G	0.123305	0.123306
A	0.1233056	0.1233057
T	0.12330567	0.12330568
E	0.123305678	0.123305679
S	0.1233056789	0.1233056789

Table 5: Performing the encoding procedure

Thus "Bill Gates" would be coded as 0.1233056789 (Table 5).

It is easy to decode this number back to bill gates by following a procedure that is the reverse of the technique mentioned above. Arithmetic coding has also been applied for Text compression (Witten et al, 1987).

Prediction by Partial Matching (PPM)

PPM was developed in 1984 by Cleary and Witten soon after arithmetic encoding was accomplished in 1976. This was followed by a series of improvements (Moffat, 1990) resulting in the version PPMC. The algorithm uses a finite context Markov Model to predict the occurrence of the next symbol based on the occurrences of the previous symbols in the text used to train the model. In reality, the method actually blends together multiple lower order Markov models in the case when a given order is unsuitable for the prediction. Thus it recursively falls onto lower order models until the least order where every symbol occurs with a probability of $1/|\sum|$ where $\sum$ is the total number of symbols in the alphabet. This accounts for the escape probability in case a symbol was never before seen in the input text used to generate the model statistics. At its introduction, PPM was thought to be able to incorporate any order model and the predictions would improve with the increasing order. However it was later discerned that an order of "5" works best and optimally and as the order increases further, the accuracy falls owing to the escape mechanism onto lower order models. The PPM* algorithm that uses an unbound context length indicates it is better to use finite deterministic contexts for application such classification of text (Hiroyuki et al, 2005).

Though the initial applications of the PPM algorithm were to enable better compression, it has been applied to text prediction, language identification, dialect identification, language segmentation, word segmentation, text categorization etc. We intend to use PPM to perform classification of data as belonging to a specific set of named entities or otherwise in biological text.

PPM for classification

By using a precise definition of the named entities we are looking for, we can construct sets of data that are composed only of these entities and sets of data that are devoid of these entities in total. This is equivalent to distinguishing the data as belonging to different classes namely A and B (if binary classification). An intelligent guess would be that by constructing models that have different statistics of symbol occurrence we can utilize PPM and cross entropy to identify a token to belong to one of these classes. In other words, a token that resembles the model A more closely in its symbol distribution rather than model B would be attributed to belong to the class A as opposed to class B and vice versa.

However, it is very pivotal that the training sets of data used to construct the model statistics of the various classes be very representative of that class of entities and also provide a clear distinction within the classes generated. Classes of data whose symbol statistics may be close to each other may not provide a good classification result as the cross entropy would not be substantially larger for one over the other.

By using models that represent gene function statements and non gene function statements and models that represent protein-protein interactions and non interactions, we have made an attempt to study the efficacy of PPM to be applied to the realm of biological text. We find that the method performs supremely well when the definition of the named entity is precise and non ambiguous. As we move into semantic interpolations and mining based on meaning and import, we notice that the performance falls markedly.

Proteins and Protein Protein Interactions

Since protein-protein interactions do not have a very strong defining principle, it required careful manual annotation and analysis of the training and testing corpora and the results achieved. Protein-protein interactions refer to the association of protein molecules and the study of these associations from the perspective of biochemistry, signal transduction and networks (Wikipedia). Proteins are the building blocks of life systems. Though it is the genes that code for proteins, it is the proteins that actually render the final function to a cell or its members. Techniques such as co-immuno-precipitation, FRET, yeast two hybrid assay, tandem affinity purification have all contributed to the ability to study protein-protein interactions on a massive scale.

Proteins are of various forms and structure and, in fact, even their functions are an outcome of their structure. It is through a complex interplay of the conformational compatibilities that multiple proteins are able to interact with each other and with other biological molecules. A simple, elegant example is that of the interactions between enzymes, a class of proteins, and their substrates. We are well aware of the lock and key and induced fit models which explain how enzymes and substrates interact by virtue of their conformational fit. It is impossible to even perceive of the biochemical processes that enzymes catalyze within cells, if proteins were not to interact.

A second direct outcome of protein-protein interactions is that of signal transduction. The responses that a cell and its components generate to an external stimuli or and environmental change is all made possible through signal transduction mediated by a cascade of protein interactions triggered by receptors present on the cell exterior. The release of second messengers, the production of antibodies conferring immunity, cell signaling, everything is a far reaching consequence of interactions among protein molecules.

Also, it is critical to examine how proteins can form multimeric complexes by attaching with homogeneous or heterogeneous members. These multimeric forms are "active" as against the inactive non multimeric forms. We also have scenarios where proteins phosphorylate other protein molecules which then act as molecular switches in a variety of biochemical processes. Post translational modification of proteins is carried by molecular chaperone proteins that interact with other proteins.

It is blatantly evident that proteins and their interactions are the very essence of the functions of the cell and hence studying these would provide a key to open a world of opportunities for therapeutic benefits.

2.3 Goals/Hypothesis

We have a set number of goals to achieve in this thesis. We aim to recognize and extract key phrases comprising named entities from biomedical corpora. We look to facilitate automated information annotation and enable advanced database curation. Studying text mining methods suited for biomedical text and establishing the efficiency of prediction by partial matching in this realm is also our focus.

3. METHODS

The Prediction by Partial Matching (PPM) text compression scheme has consistently set the standards in lossless compression of text since it was originally published in 1984 by (Cleary et al, 1984).

Compression methods encode a text string according to a given model. Formally, the average number of bits per symbol to encode a text string T can be considered to be the cross entropy H (T, P_m, L) with respect to the model Pm and the language L and is given by:

$$H (T, P_m, L) = -1/n \log_p T$$

This can be applied as a chain rule to extend the formula to include multiple contexts in the PPM family of models.

PPM uses the Markov model principle to perform the compression and the prediction. Each symbol is encoded based on the probability of the symbol calculated in the context of the 'n' previous symbols. The value 'n' acts as the order of the model. An order of 5 has been found to work optimally and higher order models do not improve the performance much (Cleary, Teahan, Witten, 1990). PPM, in reality, performs a blending of different order models through the process of exclusion to remove lower order predictions for a given symbol. This is achieved by calculating probabilities for all orders lower than the highest order specified. Most importantly a 0 order distribution is calculated from the unconditioned probabilities of the symbols and a -1 order model is also constructed where the probabilities of the symbols are the reciprocal of the alphabet size leading to very small but finite probability of occurrence. Each of the probabilities in all the order models are calculated by considering the frequency counts of the symbols.

The blending mechanism of PPM can be implemented at different levels depending on the token size. This can be either character based, word based or even sentence based. Compression experiments with a wide range of English (Teahan, 1998) show that word-based models consistently outperform the character based methods although the difference in performance is only a few percent. PPM itself is completely described and the two variants, following a series of recommendations by Moffat are called the PPMC and the PPMD methods. These two methods differ from PPM itself in that they use a different scheme to evaluate the escape probability when PPM switches to a lower order model to make a prediction. PPMC uses the number of novel or new symbols that were already observed in the given stream to calculate the escape probability. PPMD is a slight variation of PPMC where the probability of the escape symbol is the ratio of number of never before seen symbols to the total number of observed symbols. PPMD usually provides a small but noticeable improvement in prediction in most cases.

3.1 The Noiseless Channel Model

Knight, 1999 defines text mining as the process of automated or semi automated knowledge acquisition from linguistic sources. Most importantly, one would assume that the need to understand text to perform text mining is absolutely superficial. It has to be made possible to extract entities without having to understand the semantics involved.

The usefulness measure in text mining depends on the application – for example, it might be finding words in Chinese text, finding names and addresses in bibliographic references, or classifying text by language, authorship, or genre.

Witten et al, 1999 show text compression can alternatively be used as a key technology for text mining and classification. Frequently text mining is performed by defining dictionaries and grammars and progressively mapping sentences to the desired framework that best represents the entities to be identified. Witten et al use supervised training to detect sub languages of text instead of explicit programming. Language modeling techniques based on the text compression scheme PPM are used to extract meaningful low level information about the location of semantic tokens such as names, email addresses, locations, URLs and dates. They report an accuracy of 89.4% for names and 75% for phone numbers identified.

The type of algorithm commonly used in natural language processing (NLP) applications is based on a statistical framework called the "noisy channel model" (Jelinek, 1985). This has as its basis a theory developed by Shannon to model a noisy communication channel. The key idea is that the target text can be considered to be a corruption of the source text. Here, the application is formulated as a communication process- a message sent down the communication channel by a sender and the receiver tries to recover the original message that is received. This is usually done by finding the target text with maximum probability given the observed text. The process of correcting the text is often referred to as decoding. This is usually performed using a dynamic programming search algorithm called the Viterbi, 1967 algorithm.

The noisy channel model may seem a rather arbitrary and contentious characterization for NLP but leads to more robust and accurate performance than other methods. For our purposes, where we wish to exploit language models based on the state of the art text compression techniques which have been found to be highly effective in many text mining and NLP applications we wish to alter the perspective of the noisy channel model statistical framework. Rather than characterize the process as one of noisy communication- and therefore requiring a decoding process to recover the original message- our approach is to think of the process(perhaps more naturally) as noiseless communication and the emphasis is on encoding rather than decoding. Here the sender will first find the best encoding of the target message so that it can be sent with lossless property to the receiver down the noiseless communication channel. In this case the decoding process is of secondary importance mainly to verify the encoding process the emphasis instead is placed on how efficiently the target message can be encoded. Also encoded is additional information on how to transform the source message into the target message. In this case, the search process that of finding the best transformation occurs prior to the encoding phase rather than during the decoding phase.

Witten et al apply this approach to text mining. They show how a broad range of patterns such as names, email addresses, and URLs can be located in text. With their approach, training data for the model of the target text contains text that is already marked up in some order. A correction process is then performed to recover the most probable target text from the unmarked up source text. Yeates and Witten, 2000 show how many text mining applications can be recast as tag insertion problems- that of reinserting tags back into a sequence to reveal the meta–data implicit in it. Some sample applications they explore are word segmentation and identifying acronyms in text (Yeates, Bainbridge, Witten, 2000).

Teahan and Harper, 2001 have shown that by modifying PPM and enabling the algorithm to insert tags in the source text we can identify clearly all symbol locations where the method makes a shift between the models that can be either static or dynamic. This would be highly useful as it allows the selective advantage of recovering the text just by removing the tags off. Also, it indicates that the tagged regions are the regions of interest where the stream of symbols are marked up based on the statistical patterns in the training data. This tag insertion itself is implemented by using the Text Mining Toolkit (TMT) a toolkit for modeling sequential text based on text compression schemes.

3.2 Prediction by Partial Matching for Language Identification

One of the first examples of the application of PPM for classification came when it was tested for the ability to identify a text as belonging to a language. The goal is to apply PPM to generate different models for different languages (say French, German, Latin, and English) based on training text that contains symbols from each of these languages respectively. As a first principle, we will thus end with four models one for each of the languages with each model having a specific number of bits measuring the entropy of that model.

In order to check the efficiency and accuracy of the models and PPM itself as a tool for classification, the input stream of symbols is then encoded using the four different statistical models generated. This would give rise to what is known as cross entropy values i.e. the entropy of the input text with respect to each of the four models constructed from the training text.

If the input stream has characteristics/statistical patterns that resemble closely with any of the four models generated, the cross entropy of the given stream with respect to that model would be very low. As a result the model that was used to generate the least cross entropy would be the model that would closely resemble the symbol statistics in the input stream. This would enable us to conclude that the given input stream is more associated with one model against the other three. The key here is that even if the input stream were from a language that was not present in the four models generated; say the input stream was from a language like Hindi, we would still ascribe the input stream to belong to one of the four classes that were present. The following figure (Figure 2) indicates the cross entropy principle for classification.

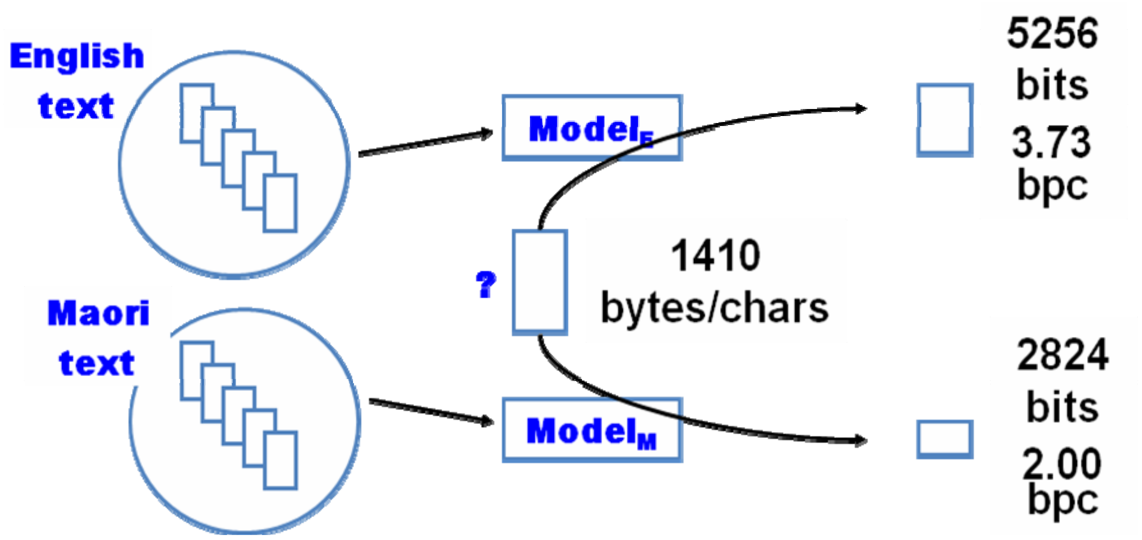


Figure 2: Cross entropy principle for a binary classification (Teahan, ppt for TMT)

In the above picture, we represent two models one trained from English Text and another from Maori[1] text namely Model E and Model M. We then compress our input text to calculate the cross entropy with respect to the two models to be 3.73 bits per character and 2.00 bits per character for English and Maori respectively. Therefore we would then conclude that our text belongs closely with the Maori language than with English.

Therefore we need to understand some salient features.

[1] The language of the original people of New Zealand

1. We are using a relative means to establish that our input stream belongs to one class of objects than another but only within the set of classes for which models have been generated and a cross entropy is available.
2. To achieve a good classification, we must be sure to include the classes to which the input stream may closely be associated with. There is no way to identify, for example, that Hindi is not really any of English, German, Latin or French, if we did not realize the importance of our training data generating the models.
3. We also need to ensure that the training sets are sufficiently large and provide a very good difference in the cross entropies achieved. In other words, greater the difference in the cross entropy values for the models, greater the chance that our input stream belongs to the class with least cross entropy.

The following table (Table 6) illustrates (Teahan) tests on language identification based on PPM.

Training/testing	**English**	**French**	**German**	**Latin**
English	**1.56**	2.30	2.38	2.33
French	2.54	**1.71**	2.56	2.59
German	2.64	2.60	**1.75**	2.59
Latin	2.87	2.92	2.91	**1.97**

Table 6: Results of testing PPM over language classification (Teahan, ppt for TMT)

We can observe that the cross entropy values along the main diagonal are the least for the given row indicating that PPM classifies the text to its correct class based on the approach of cross entropy.

3.3 Implementing PPM: A simple example

PPM clearly works well to identify text as belonging to a particular class based on cross entropy. We will examine using a simple example as to how PPM uses the blending of finite context Markov models of various orders to generate the symbol probability and predict the occurrence of the next symbol in the stream.

In effect the approach of blending lower order Markov models easily removes an arbitrary pseudo count being used in case a character that was never before seen in the input stream appears. PPM uses the previous “k” symbols to calculate the probability of occurrence of the next symbol. To achieve this, the algorithm stores all length-k subsequences of characters and calculates a probability distribution of these subsequences for different values of k. The algorithm follows a recursive procedure to determine the highest order K which can be used to perform arithmetic coding and generate a probability for the current symbol. Thus it contiguously switches from order k to lower orders until an order which is most suitable for prediction is reached. The test of suitability is then to be studied. In case a character that appears in the stream is novel to order k then it cannot be assigned a probability with a model of that order. Hence the k^{th} order model becomes unsuitable and the next lower order model is chosen. In the case that even this model is unsuitable i.e. the character is novel to this model also, then the next lower model is triggered etc. To ensure that the recursion does not proceed indefinitely, a final model is present that contains all the characters in the given alphabet. Each time the decoder switches from one higher order model to a lower order model it generates an escape sequence with a specific probability calculated for that model. Thus the algorithm makes sure that all characters are assigned a probability value and also avoid the use of arbitrary pseudo counts.

Let us consider an input text “abracadabra”. We shall now observe how PPM will use the symbol frequencies to generate models of order k=2, 1, 0, and -1 and use these distributions to predict the probability of occurrence of the next symbol in the context.

The following table (Table 7) describes how PPM generates its probability distributions.

Order k = 2			**Order k = 1**			**Order k = 0**			**Order k = -1**		
Prediction	**c**	**p**	**Prediction**	**c**	**p**	**Prediction**	**c**	**p**	**Prediction**	**c**	**p**
ab →r →esc	 2 1	 2/3 1/3	a →b →c →d →esc	 2 1 1 3	 2/7 1/7 1/7 3/7	→a	5	5/16	→A	1	1/|A|
ac →a →esc	 1 1	 1/2 1/2	b →r →esc	 2 1	 2/3 1/3	→b	2	2/16			
ad →a →esc	 1 1	 1/2 1/2	c →a →esc	 1 1	 1/2 ½	→c	1	1/16			
ba →a →esc	 2 1	 2/3 1/3	d →a →esc	 1 1	 1/2 1/2	→d	1	1/16			
ca →d →esc	 1 1	 1/2 1/2	r →a →esc	 2 1	 2/3 1/3	→r	2	2/16			
da →b →esc	 1 1	 1/2 1/2				→esc	5	5/16			
ra →c →esc	 1 1	 1/2 1/2									

Table 7: Running example for input stream "Abracadabra" (Cleary et al, 1995)

The above table is generated by first enumerating the different length-k subsequence strings in the input. With k=2 we get 7 different strings as in the first column of the table. With order k=1 we get 5 strings etc. Each column has an escape sequence whose count is equal to the number of "distinct" symbols seen in the input until then. Thus the third column of order k=0 has a count of 5 associated with the escape sequence as 5 character were observed in that order. The final column has the lowest order model that would be used if a character was totally unseen.

Consider three characters which may appear following the input symbols "abracadabra". The probabilities of occurrence of these characters are shown in (Table 8) below.

Character	**Probabilities (without exclusion)**	**Probabilities (with exclusion)**	**Code space occupied**
c	1/2	1/2	$-\log_2(1/2)$=1bit
d	1/2*1/7	1/2*1/6	$-\log_2(1/14)$=3.6bits
t	1/2*3/7*5/16*1/|A|	1/2*3/6*5/12*1/|A|-5	$-\log_2(1/3822.33)$=11.9bits

Table 8: Probability calculation for subsequent characters

If the next character in the stream is 'c' then it matches the first column with k=2 where 'c' followed the context "ra" previously. Therefore 'c' would be encoded with a probability of ½ or 1 bit.

Consider the character 'd' appearing following "abracadabra". We can see that'd' never followed the characters "ra" previously. Therefore the k=2 order model becomes unsuitable and an escape sequence is generated with a probability of ½. Following this the next lower order i.e. k=1 model is chosen and it can be seen that'd' followed 'a' with a probability of 1/7. Therefore the combined probability for 'd' to appear is the product of the two probabilities and equals 1/14 requiring 3.6 bits to encode this symbol.

Finally, let us consider the case where a symbol was never seen in the text stream. Consider the character 't'. Since 't' did not occur at all in the input stream it would generate an escape sequence for all models irrespective of k. However it will finally end with k=-1 where the probability of finding 't' would be 1/|A| or equal to 1/|256|. By multiplying with escape sequence probabilities of all models prior to this, we see that‘t’ has a probability of 1/3822.33 and so requires 11.9 bits- a very high value (Cleary et al, 1995).

3.4 Gene Function Identification

The transcription and translation of DNA into RNA and proteins respectively constitutes the central dogma of life. The small units of heredity within this long double stranded DNA helix are the genes. As genes and their products interact with one another, the organism develops its phenotype in a specific manner. Therefore an understanding of genes and their corresponding functions is a very significant portion of research in biotechnology and medicine. By understanding the functions of a gene, one can theoretically, target these genes to modify, alter or sometime even knock out the function for numerous therapeutic and medical benefits. Such an advantage has led researchers to devise high throughput technology such as the Yeast Two Hybrid System and Microarray chips to facilitate the simultaneous study of thousands of genes from any given cell of any given organism at any given time and condition.

The direct outcome of this biotechnology revolution has been a tremendous overflow of gene function information with the concomitant increase of the articles being submitted to literature databases such as the PubMed. As more and more articles focusing on genes, their functions and methods to modify them were being published, an increasing number of researchers using these articles to identify prior information on their gene of interest, or to identify medical ramifications of existing functions or even to identify putative drug targets rose considerably. The number of searches being performed on these journal articles based on genes grew exponentially between 1997 and 2007 (NCBI). To handle such voluminous information and to efficiently allow searches on the articles thereby reducing the time and effort involved in locating articles of interest, most public databases began to curate these articles to index them based on key terms such as gene names, gene functions etc.

One such effort is the ongoing effort of the National Center for Biotechnology Information (NCBI) to construct a database “the locuslink” (now superseded by Entrez Gene) that would encompass the information annotation for a given gene as derived from journal articles. In particular Locuslink has a field that includes the function of the gene being studied. This is to allow researchers to retrieve information for the gene function from the database.

Constructing such databases that would facilitate proper gene annotation and reduce the time involved in identifying them needs the effort of multiple human curators who need to go through each article as it gets published and extract the necessary pieces of information. This process is laborious, error prone and vulnerable to variability in semantic interpretations from one curator to another. With the availability of the electronic information storage system and a growing interest in data mining and named entity recognition, a possible solution that could alleviate the problems of human curation would be to identify named entities in biological corpora. In particular, an effective solution to the problem of gene function identification would prove very beneficial in terms of reducing the human intervention in annotation but also providing a first means to further entity identification. The gene functions that may result from such an effort could be further used to improve search terms for information retrieval. Also, the possibility that an automated approach could identify new relations that a manual curator previously did not identify is a very realistic one. The final applications seem amazingly overwhelming in that this method could be used by scientists in the medical domain to extract genes that could be drug targets depending on the function they perform.

To facilitate gene annotations further, NCBI constructed the GeneRIF dataset that comprehensively covers the gene functions for the known genes. GeneRIF in itself stands for Gene Reference Into Function. GeneRIFs are always associated with specific entries in the Entrez Gene database. The records in GeneRIF comprise of the PubMed identifier of a scientific publication from which the evidence for the Gene function was obtained. Most GeneRIF statements also appear directly taken from the title of the abstract of the scientific publication. GeneRIF statements are generated by indexers at NCBI. However, users may also submit their own GeneRIF annotations provided a valid Gene ID exists for the given gene in the Entrez gene database. Here are some GeneRIFs taken from Entrez Gene for GeneID 7157, the human gene TP53.

- p53 and c-erbB-2 may have independent role in carcinogenesis of gall bladder cancer
- Degradation of endogenous HIPK2 depends on the presence of a functional p53 protein.

- p53 codon 72 alleles influence the response to anticancer drugs in cells from aged people by regulating the cell cycle inhibitor p21WAF1
- Logistic regression analysis showed p53 and COX-2 as dependent predictors in pancreatic carcinogenesis, and a reciprocal relationship to neoplastic progression between p53 and COX-2. (NCBI)

Our approach to identifying Gene function phrases makes use of the Prediction by Partial Matching algorithm for text compression to classify a token as belonging to gene function class or a non gene function class and further markup the token within the input text. Therefore the problem is a binary classification problem to be solved using PPM. The goal, then, is to collect data representative of gene functions and non gene functions to generate two models by using the PPM text compression scheme. Finally, we can expect to classify the tokens (words, phrases, sentences etc…) to belong to one of the two classes based on the cross entropy approach outlined previously. To construct the model for Gene function statements, we chose GeneRIF to be our source, naturally, and to represent the non gene functions, we have used a number of different approaches to study the system performance. The actual PPM algorithm has been implemented as a toolkit using the C language, called as the Text Mining Toolkit (TMT) available at
http://www.aiia.cs.bangor.ac.uk/TMT

3.4.1 GeneRIF for Training

Since GeneRIF contains a comprehensive set of PubMed identifiers of documents and the evidence text of Gene Function statements, it was naturally the choice to be used to pick out Gene Functions for training the TMT to generate a Gene Function model. To obtain a clean training set, the GeneRIF source was processed to remove the PubMed identifiers leaving behind only a collection of Gene Function statements. Using this as our training set for Gene Function model, we utilized the TMT to generate model statistics for the same. While performing the training we set aside 200 records randomly for testing the model.

3.4.2 Non Gene Function Text for Training

The first model having generated, the question is next to generate the non gene function model that can be used to well distinguish the Gene Functions from the non gene functions in a given abstract of text. To achieve this, several variants were used along with a thorough monitoring of the system performance. In general, the idea is to remove any gene function phrases off the abstracts and then use the rest of the abstract phrases, which are the non gene functions, to obtain the second model. Therefore, one has to first identify all the abstracts which provide the evidence text for GeneRIF, through the PubMed identifier present in the GeneRIF source. Then a match is executed to locate the gene function statement within the abstract that actually GeneRIF uses as the evidence text. This matching is not trivial because GeneRIF evidence text does not have an EXACT match with the corresponding statement in the actual abstract. It was identified that an exact match occurred only 7.48% of the time further emphasizing our need to slowly vary the non gene function model until saturation in the performance is achieved. Once a corresponding phrase is located, the entire abstract without this phrase is then appended to the training text required to construct the model for the non gene functions. Another rationale behind such a procedure is that the sentence framework within biological corpora may be different from the general language framework but significantly characteristic of statements that do not talk about genes and their functions yet are part of the abstract itself. We noticed that this naïve model worked suitably efficiently (better than random guessing) but far less accurate than other variants discussed further.

Both the Gene Function and non gene function models were constructed using TMT with an order of “5” as it was shown to be the most optimal. Also a character based processing was chosen because in general, Biological terms make it difficult to determine word boundaries. Specifically, when the terms involve gene names or protein names that could be one of several different forms in each text, depending on the style of the author’s writing, the boundary of words become difficult to ascertain.

3.4.3 Testing

To avoid a biased testing by including the abstracts in training to be a part of the test set, which would skew the results considerably, we left out 200 abstracts and corresponding GeneRIF gene functions from the GeneRIF source even during the training phase. A random set of abstracts from this set of 200 formed the test set for the variants (modifications to the training set based on performance analysis of the current model) used in the identification of Gene functions. The testing is carried out by using the same parameters as for training using TMT. The difference and the special feature of TMT is that first it identifies phrases to belong to either gene functions or non gene functions and then in the second step it marks up these phrases with appropriate tags specified during the training.

Therefore once TMT marked up gene functions within the abstracts, these were fished out selectively using PERL scripts and then compared with the actual GeneRIF representative for that abstract. One has to notice that there could be multiple gene functions arising out of TMT markup and one has to locate one true representative out of these and then finally compare this representative with the actual GeneRIF representative.

3.4.4 Variants

Variant #2

This variant was different from the first one in that the first one used only those abstracts for training, that had an exact match between the GeneRIF representative and the sentence in the abstract. Therefore many gene functions which did not find an exact match were left off the training in the first case. However in the second variant, even abstracts that had partial matches of a sentence with a GeneRIF representative were chosen for the training. But we ensured that the complete sentence which had a partial match did not appear in the training data for the non gene function model.

Variant #3

The third variant was composed to improve the distinguishing ability of TMT between gene functions and non gene functions. In this variant, the non gene function model was enhanced by appending the set of sentences from the King James' bible data collection available as a part of the TMT. This was because of the observation that some of the false positives from variant #2 were actually English sentences which did not have gene function mentions.

Variant #4

By keeping the non gene function training set as is from Variant #3, we changed the Gene Function training set by expanding it to include all the Gene Function statements from GeneRIF irrespective of whether they had a partial match with the abstract or not. This not only expanded our Gene function repertoire but also improved the performance of the system well. Again, we had removed 200 abstracts from the entire set prior to training to be used for testing purposes.

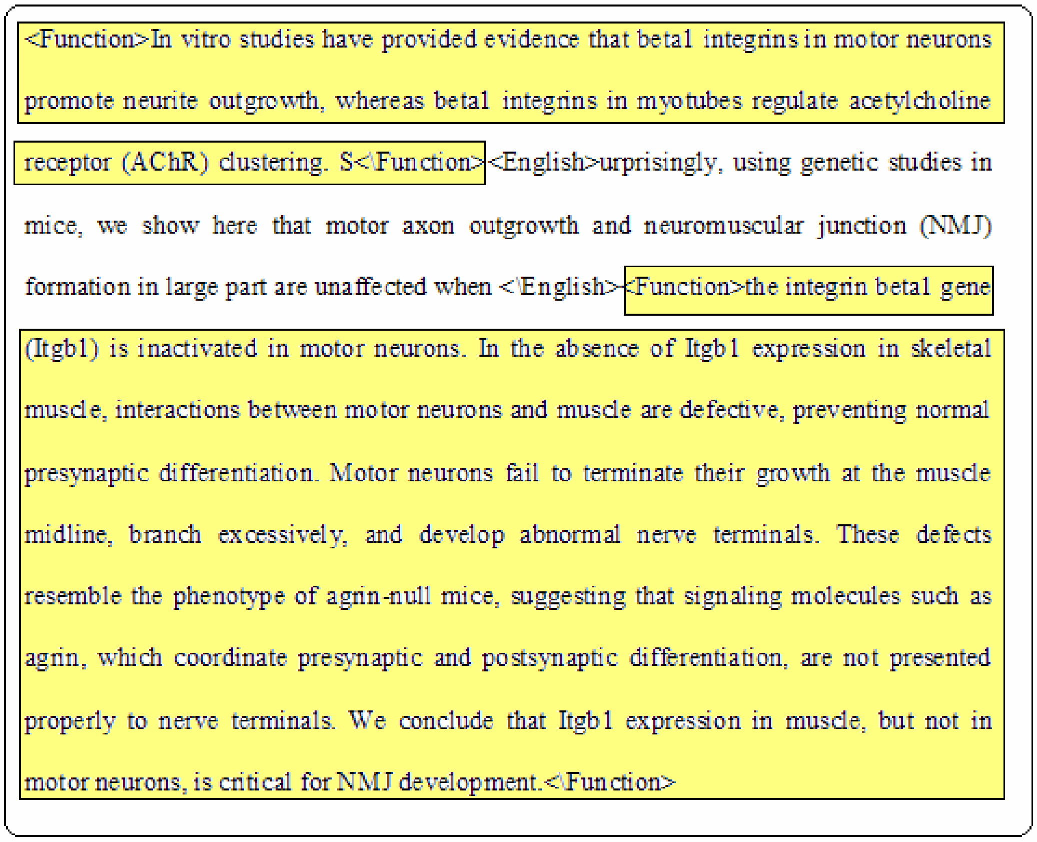
Variant #5

The final variant in the series had a post processing phase associated with the final results of variant #4. This was carried to efficiently identify the best possible representative from the set of multiple putative functions marked by TMT. This was carried out by using a distance measure to compare the number of word matches between the gene function marked and the title of the abstract. As mentioned earlier, the GeneRIF source contains representatives frequently isolated from the title of the abstract. Therefore, following this we used that function statement that captured the title the best.

Along with the five variants proposed above, a second task was considered to compare how the system performed in general without relation to GeneRIF representatives. The second subtask performed using the same approach was to restrict the comparison with

GeneRIF functions and to directly evaluate all the candidate gene functions manually to ascertain whether they were actually gene functions are not. Due to the nature of the second task, it eliminated the need to identify a single representative out of the candidates from any given abstract and so the fifth variant with the post processing was thwarted for this task. However all the evaluations were performed again to ensure there was no bias in the system accuracy measurement.

A typical markup example after applying the TMT on the abstract to identify gene function is shown (Figure 3).



<Function>In vitro studies have provided evidence that beta1 integrins in motor neurons promote neurite outgrowth, whereas beta1 integrins in myotubes regulate acetylcholine receptor (AChR) clustering. S<\Function><English>urprisingly, using genetic studies in mice, we show here that motor axon outgrowth and neuromuscular junction (NMJ) formation in large part are unaffected when <\English><Function>the integrin beta1 gene (Itgb1) is inactivated in motor neurons. In the absence of Itgb1 expression in skeletal muscle, interactions between motor neurons and muscle are defective, preventing normal presynaptic differentiation. Motor neurons fail to terminate their growth at the muscle midline, branch excessively, and develop abnormal nerve terminals. These defects resemble the phenotype of agrin-null mice, suggesting that signaling molecules such as agrin, which coordinate presynaptic and postsynaptic differentiation, are not presented properly to nerve terminals. We conclude that Itgb1 expression in muscle, but not in motor neurons, is critical for NMJ development.<\Function>

Figure 3: Example Markup of an abstract by TMT

The region of the abstract marked in yellow is the set of statements identified by TMT as gene functions.

The TITLE of the abstract is (Figure 4)

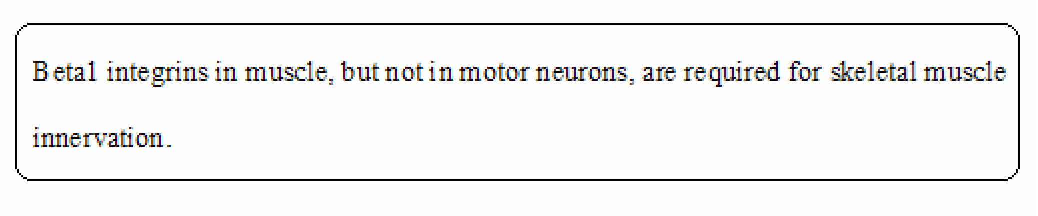

Beta1 integrins in muscle, but not in motor neurons, are required for skeletal muscle innervation.

Figure 4: Title of the abstract for test

Match scores for function 1 with title is 5 whereas that for the second one is 4.

Post processing yields the first function marked in the abstract and this is the actual GeneRIF representative.

3.5 Protein-Protein Interaction identification

With the successful levels of performance achieved on the gene function identification objective, the next phase to apply the technique to a problem that is a bit fuzzier by definition as compared to gene functions. The task was to identify protein-protein interactions in the biomedical corpora using an approach quite similar to the one employed for discerning gene functions from the same text.

Our overall objective remains unchanged in that we are looking to extract protein-protein interaction phrases from biomedical corpora facilitating an easy annotation scheme and also allowing researchers to identify these valuable entities to construct interaction networks and proceed to identify drug targets.

Our focus, alternatively, is to determine, how effectively PPM based compression can contribute to the same and what differences might arise due to the nature of the problem and the entity being tackled. We have proceeded along similar lines to gene function identification in that; we have constructed two training sets each representing the class to which we would like to classify our input stream. These data sets compose of manually curated protein interaction phrases obtained from the intact database and the non protein interaction sentences were extracted from Brown's English corpus frequently used for NLP.

3.5.1 Training

Many different sources were considered to compose compact and sufficiently large training set for protein interaction identification. There are multiple approaches that were scrutinized. We could create a dictionary of protein names first to identify all occurrences of protein names within the article. We could then use a dictionary of interaction terms such as "binds to", "phosphorylates" etc. to identify all sentences that contain a protein name as well as the interaction term. Alternatively, we could identify protein-protein interaction statements/phrases directly by constructing a dictionary that has both the protein names and their interactions together as single entities.

The first method has the disadvantage in it that there are protein interactions within biomedical articles that do not actually have protein names associated in the same sentence. The protein name would have been mentioned in the beginning of the article and a preposition would be used when talking about its interaction. This leaves us with the possibility of missing vital interactions from articles and the inability to discover new relations. The second approach is comfortably away from such a disadvantage and subsequently, we decided to use this approach. The problem however is to identify a suitable source that can provide us with such interaction sentences and phrases. After a thorough examination of a multitude of data sources available from the (BioCreative) consortium, we decided to make use of the data mining archive provided by the (IntAct) database. Some of the sources that were studied included the Yapex corpus, the Plake dataset used in the (LLL05) contest, the datasets from Mint, BIND, DIP all of which showed performances considerably lesser than the random guess attributed to the textual extracts that contain an interaction but are highly non specific.

The Intact data archive for protein-protein interactions is a tab delimited list of PubMed identifiers and the evidence protein interactions obtained from the articles corresponding to this identifier. The direct utilization of the IntAct data source culminated in a higher performance (upto 44%) but was still an issue because the data was more of a collection of evidence excerpts than concise evidence sentences or phrases. Our answer to this challenge was to create a manual annotation of the evidence text from IntAct data source and utilize this concise data set for training and generating a model. About 1169 interaction sentences were isolated from the IntAct data source and manually refined to form our training data. Each of the excerpts that formed the evidence text for Intact data archive were scrutinized to filter out any common English constructs that were present apart from direct interaction phrases (Figure 8). Also included were sentences that were completely interaction based and did not include non interaction key words. Simultaneously we set aside 100 abstracts from the same intact source to act as our test set. The test set had zero overlap with the phrases used for training the TMT.

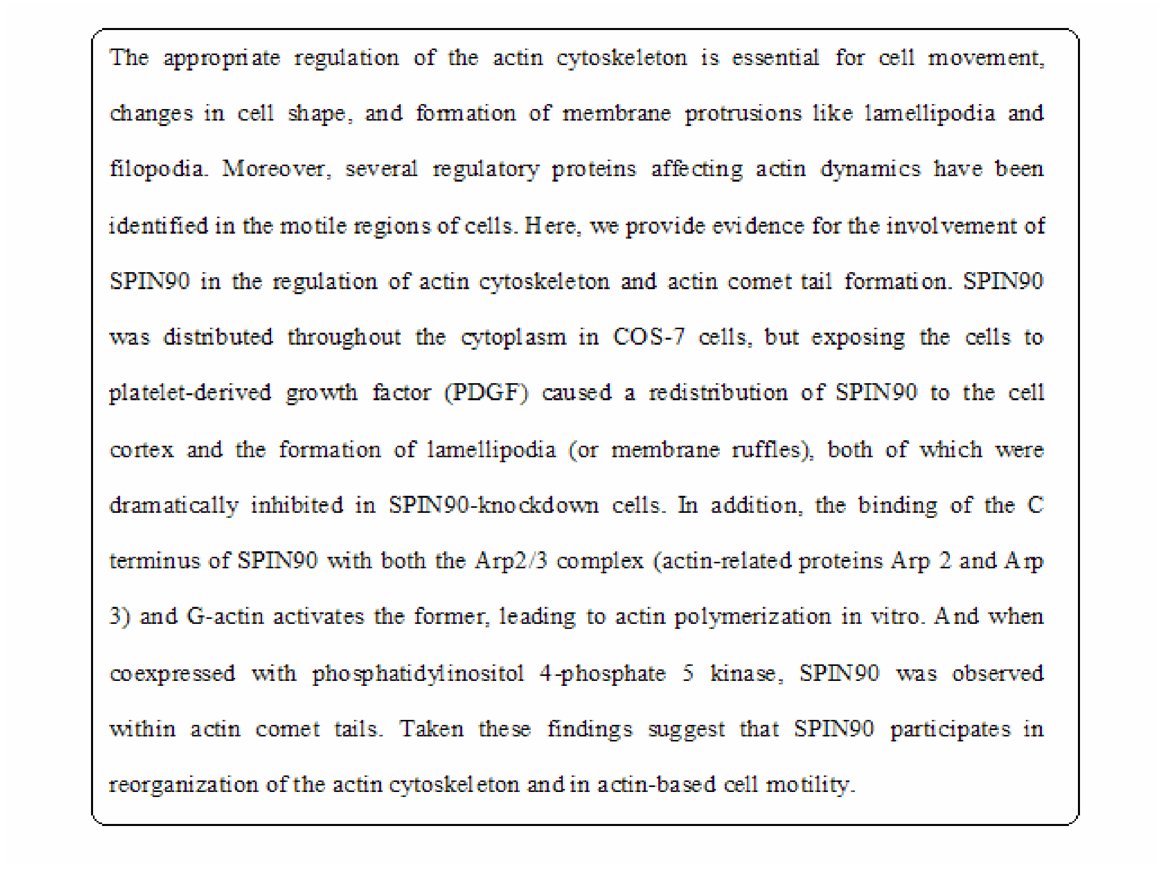
3.5.2 Testing

The efficiency of PPM to identify protein interaction phrases was tested on 100 abstracts chosen randomly from the IntAct data archive. These 100 abstracts did not form part of the training set that was obtained by refining the IntAct data archive. The goal was to identify all protein interaction phrases within the 100 abstracts and then measure the precision and recall of the system. To create the reference list of interactions from the 100 test abstracts thereby achieving a benchmark to compare the TMT marked phrases with, the 100 abstracts were manually annotated to unearth all protein interaction phrases within these abstracts. TMT would be applied to the same 100 abstracts and the phrase marked up by TMT as interaction fragments would be compared with the manually annotated dataset for an overlap. The evaluation of an overlap and the definition of a true positive were critical. Our evaluation scheme was straight forward as we chose to consider a three word match between a TMT marked phrase and a manually annotated phrase. The three words were not to be articles or prepositions so that we can remove spurious overlaps that were meaningless. Also critical was the use of a sentence boundary. Gene functions are generally complete in the sentence form and do not exist as incomplete phrases unlike protein interactions. To overcome this issue, we chose to consider a sentence to end at a period and also to begin at a period. For example, if the manual annotation of a scientific abstract for protein interactions had two sentences as

"A phosphorylates B. B was found to be bound to C." then we used "A phosphorylates B" as one sentence and "B was found to be bound to C" as a second sentence. Since TMT markup never yielded a unit longer than a sentence (a markup can cut across a sentence boundary to span across two sentences at the maximum), it did not require a tedious break down. However, even if TMT marked entities as "<ppi>phosphorylates B. B was found to be bound</ppi>" then, "phosphorylates B" would be one phrase and "B was found to be bound" was another separate phrase. This ensures that our counts for measurement of accuracy were not flawed and the numbers had a certain degree of credibility.

The datasets are available at: http://mypage.iu.edu/~mmahoui/data-set-01-04-08/

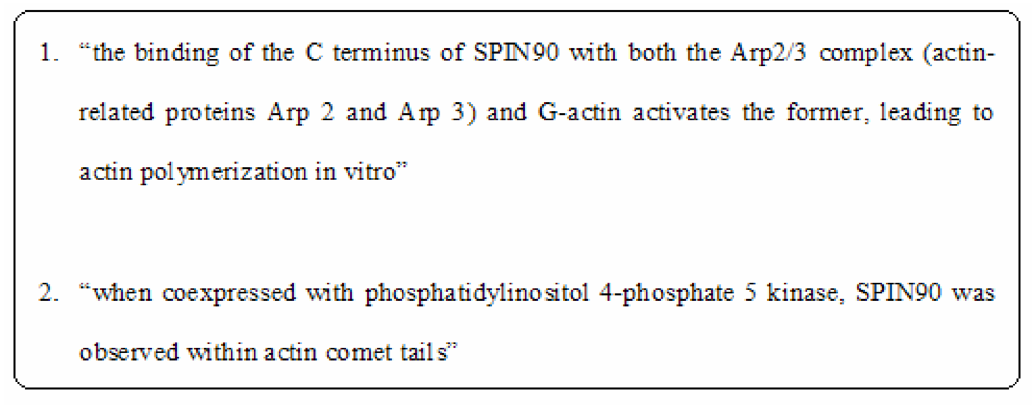
As an example let us consider (Figure 5), PubMed Identifier 16253999



The appropriate regulation of the actin cytoskeleton is essential for cell movement, changes in cell shape, and formation of membrane protrusions like lamellipodia and filopodia. Moreover, several regulatory proteins affecting actin dynamics have been identified in the motile regions of cells. Here, we provide evidence for the involvement of SPIN90 in the regulation of actin cytoskeleton and actin comet tail formation. SPIN90 was distributed throughout the cytoplasm in COS-7 cells, but exposing the cells to platelet-derived growth factor (PDGF) caused a redistribution of SPIN90 to the cell cortex and the formation of lamellipodia (or membrane ruffles), both of which were dramatically inhibited in SPIN90-knockdown cells. In addition, the binding of the C terminus of SPIN90 with both the Arp2/3 complex (actin-related proteins Arp 2 and Arp 3) and G-actin activates the former, leading to actin polymerization in vitro. And when coexpressed with phosphatidylinositol 4-phosphate 5 kinase, SPIN90 was observed within actin comet tails. Taken these findings suggest that SPIN90 participates in reorganization of the actin cytoskeleton and in actin-based cell motility.

Figure 5: Example abstract for markup with TMT

The manually annotated interaction sentences are in (Figure 6)

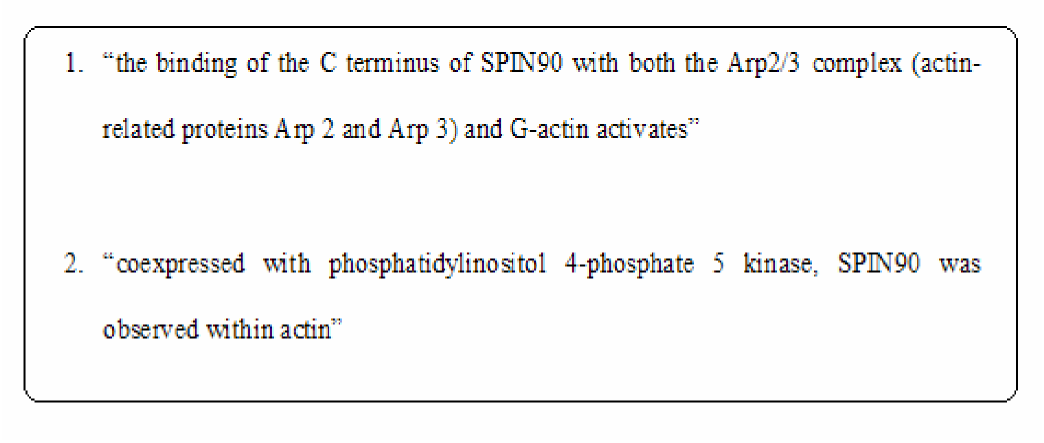


1. "the binding of the C terminus of SPIN90 with both the Arp2/3 complex (actin-related proteins Arp 2 and Arp 3) and G-actin activates the former, leading to actin polymerization in vitro"

2. "when coexpressed with phosphatidylinositol 4-phosphate 5 kinase, SPIN90 was observed within actin comet tails"

Figure 6: Results of manual annotation

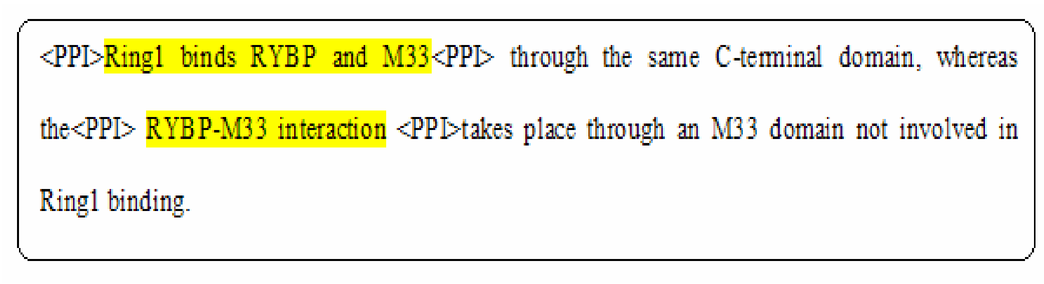
The TMT marked interaction phrases are as in (Figure 7)



1. "the binding of the C terminus of SPIN90 with both the Arp2/3 complex (actin-related proteins Arp 2 and Arp 3) and G-actin activates"

2. "coexpressed with phosphatidylinositol 4-phosphate 5 kinase, SPIN90 was observed within actin"

Figure 7: Results of TMT markup



<PPI>Ring1 binds RYBP and M33<PPI> through the same C-terminal domain, whereas the<PPI> RYBP-M33 interaction <PPI>takes place through an M33 domain not involved in Ring1 binding.

Figure 8: Sample manual annotation of Intact PPI, PMID 10369680

4. RESULTS

The results are presented in two sections. The first section provides the data obtained from the gene function identification test and the second section depicts the results from the protein-protein interaction identification experiments.

4.1 Gene Function Identification

There are two tasks associated with the Gene function identification experiments. The first one focuses on the identification and selection of a single gene function representative from each of the abstracts tested while the second task is oriented to determine all the plausible gene functions from each of the articles. The results are published for each of the tasks in Table 9 and Table 10 and for each variant described earlier. The system performance is measured in terms of precision and recall where

$$\text{Precision} = \frac{\text{Gene functions that were relevant and were retrieved by TMT}}{\text{Gene functions that were retrieved in all}}$$

$$\text{Recall} = \frac{\text{Gene functions that were relevant and were retrieved by TMT}}{\text{Gene functions that was relevant in all}}$$

$$\text{F-score} = \frac{2*\text{precision}*\text{recall}}{\text{precision} + \text{recall}}$$

Task 1 GeneRIF function identification

Method	Precision(raw)	Recall(raw)	Precision(percent)	Recall(percent)
Variant #1	58/91	58/100	63.76	58.00
Variant #2	79/115	79/100	68.69	79.00
Variant #3	89/120	89/100	74.17	89.00
Variant #4	97/125	97/100	77.60	97.00
Variant #5 (post processed)	93/95	93/96	97.89	96.87

Table 9: Performance measures for the 5 variants

Method	F-score
Variant #1	0.6074
Variant #2	0.7348
Variant #3	0.8091
Variant #4	0.8622
Variant #5 (post processed)	0.9737

Table 10: F scores for the 5 variants

The precision is calculated with different denominators in each of the variants, as by definition, we need to calculate the precision for the total number of gene function phrases retrieved and not the total number of gene function phrases that are actually present. The recall however is always calculated for the total number of relevant gene functions and this turns to be 100, one for each of the 100 test abstracts sampled randomly from the 200 test records set aside. We notice that the performance improves consistently for all the variants and the best performance records an F-score of 0.9737.

Task 2 Gene function identification

Method	Precision(raw)	Recall(raw)	Precision(percent)	Recall(percent)
Variant #1	72/103	72/142	69.90	50.70
Variant #2	105/125	105/142	80.00	73.94
Variant #3	120/135	120/142	88.90	84.50
Variant #4	134/158	134/142	84.81	94.36

Table 11: Performance measures for the 4 variants

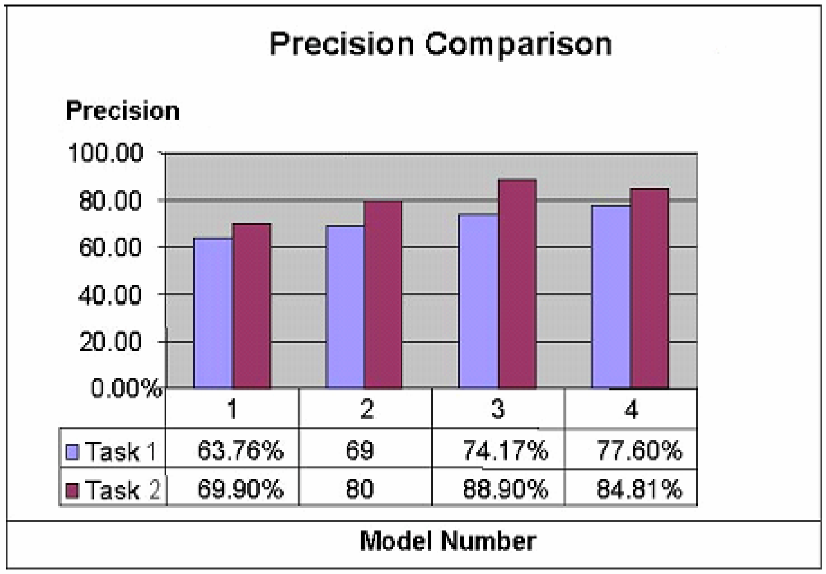
Method	F-score
Variant #1	0.5877
Variant #2	0.7685
Variant #3	0.8664
Variant #4	0.8933

Table 12: F scores for the 4 variants

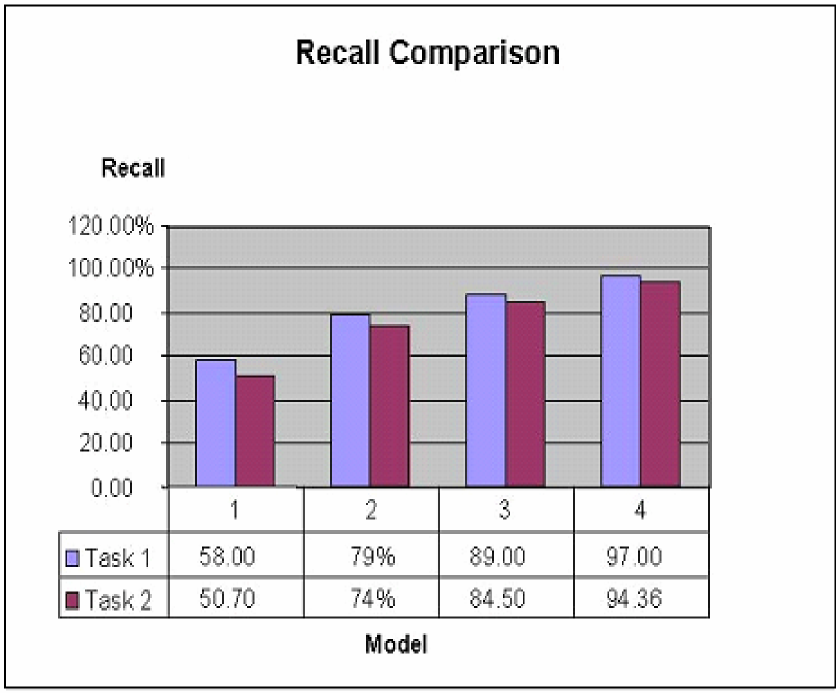
For the second task, we manually identified 42 more gene functions from within the 100 test abstracts apart from the GeneRIF representatives. By evaluating the PPM based gene function phrases against the manually annotated ones, we have obtained the foreseen results. The fifth variant is absent here as we do not have the need to select a single representative from all of the candidates. We observe again that the performance steadily improves with the system performing its best with an F-score of

0.8933.

Comparison of Precision and Recall for task 1 and task 2



	1	2	3	4
Task 1	63.76%	69	74.17%	77.60%
Task 2	69.90%	80	88.90%	84.81%



	1	2	3	4
Task 1	58.00	79%	89.00	97.00
Task 2	50.70	74%	84.50	94.36

Figure 9: Comparative analysis of Precision and Recall for Tasks 1 and 2

The application of PPM for task 2 shows an improved precision but a weakening recall as compared to that of task 1 (Table 11) (Table 12). The reason for this is, with an addition of 42 more gene functions the number for false positives from task 1 were reduced and there was an increase in the number of true positives. However, as the number of gene functions that were actually relevant increased, the corresponding false negatives also increased leaving a reduction in the recall values.

4.2 Protein-Protein Interaction Identification

We achieved a satisfactory performance of PPM when TMT was applied to the task of identifying protein–protein interactions. Since our preliminary experiments with multiple data sets performed poorly and as we learnt from our result established through tasks 1 and 2 of the Gene function identification experiment, we constructed our best performing training data and have the set of results measuring the performance of the system. However, we note that the performance can be improved by considering a post processing phase to reduce the number of false positives involved. We notice that the precision is 60.125% and the recall is 80% giving us an F-score of 0.6865. The actual matrix constructed is as below.

Measure	**True Positives**	**False Positives**	**False Negatives**
Value	288	191	72

Table 13: Performance measures for PPI identification

	Manual annotation	**PPM markup**
Number of Protein-Protein interactions	360	479

Table 14: Total counts of PPI marked

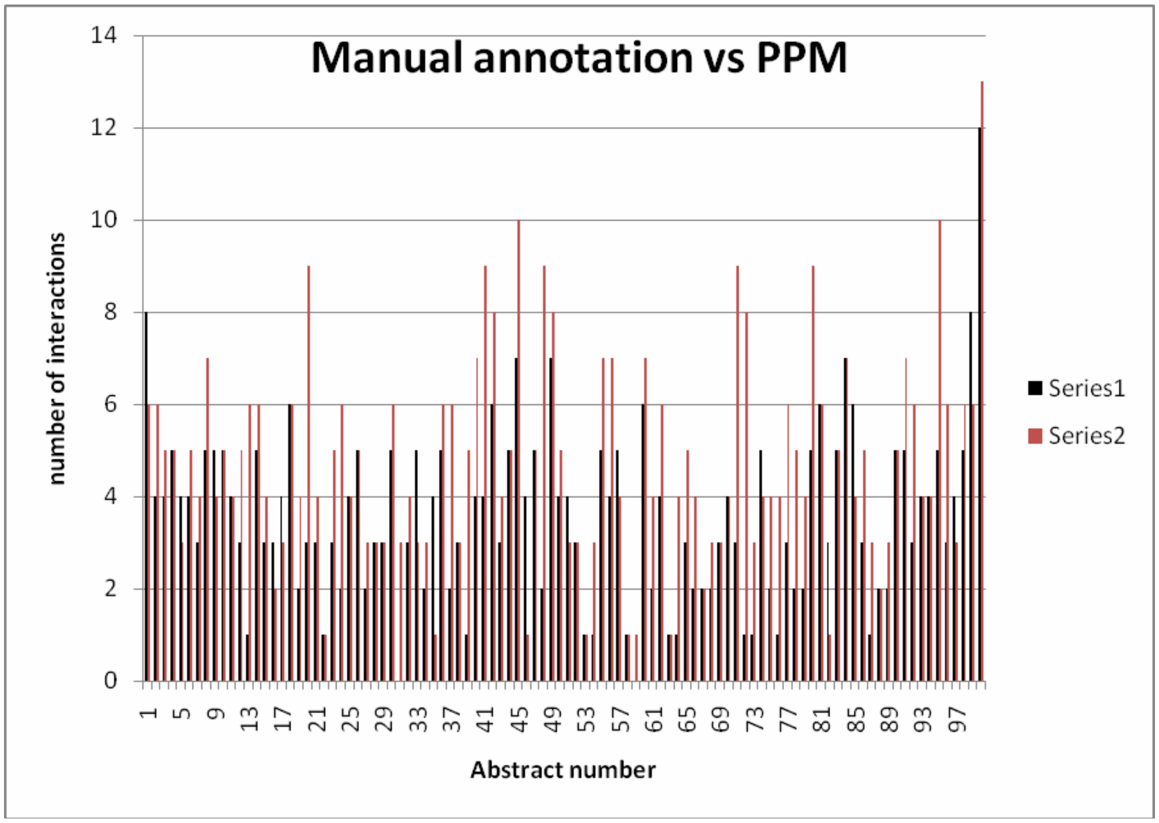


Figure 10: Comparison of number of PPI marked manually and by TMT

(Series 1- Manual annotation, Series 2- TMT annotation)

The actual comparison of the number of interactions predicted by TMT and the number of interactions obtained by manual annotation is depicted in the above bar chart.

5. DISCUSSION

5.1 Critical Analysis

In our work, we have attempted to use text compression through the prediction by partial matching algorithm for identification of named entities in biomedical corpora. We have successfully applied the technique for two different applications to better understand the nature of problems to which the text compression based classification can be usefully applied. We have chosen to extract entities such as gene functions and protein-protein interactions keeping in view the vast impact the success of the method can bring on the medical community and also not losing sight of the variability that can help us in understanding the required tuning for PPM to be perpetually used to extract any named entity from biological text. Based on our results from the experiments performed, we have charted a few conclusions and many reasons for the success and failure of PPM on different data sets. We have also let ourselves open to address a few interesting issues that can potentially improve the usefulness of PPM as an entity extractor and make it available for the scientific community at large.

The difference in performance of PPM on gene functions and protein-protein interactions indicates to us several cues to the kind of entities that can be easily extracted by this method and the kind of entities that will require a more comprehensive training set to be extracted with a similar success if not more. We note that the problems we have tried to address are fundamentally different in the precise definitions they carry. Gene function statements, on the one hand, are more clearly stated across scientific databases and archives as it is a much more explored and researched subject when compared with protein-protein interactions. On the other hand, protein-protein interactions are still in the nascent stages of research and they do not enjoy the same kind of specificity of datasets that Gene Function datasets possess. This leads directly to a reduction in the size of data that can be used to train TMT to generate a model representative of that specific class. As a result we observe a marked difference in the F-scores for the two problems. It is often believed that protein interactions should be more concise and compact as opposed to Gene functions. But this is far from what is observed. The datasets we currently have on the public domain for protein-protein interactions phrases in articles of scientific journals is not really a phrase or a sentence. We actually observe that these are more close to paragraphs of text that may have the protein interaction phrase included in them. As a result there is a tremendous loss in the precision of the system when evaluated at the phrase level. The recall in this case lies very close to a 100% as virtually the entire abstract is marked as a protein-protein interaction. In the wake of such a persistent problem, we manually curated these public archives (from Intact) to obtain a very specific set of phrases that would indicate protein-protein interactions rather than paragraphs that contain a large amount of English sentences interleaved. This would stand as an independent contribution of ours, a set of 1000 protein interaction phrases for text mining purposes. What we observed was a great increase in the precision of the system to a satisfactory 60%. However, we noticed certain features of the data set we used for training that could be a possible explanation for this behavior.

For one, we observe that interaction statements are seldom direct. We see a lot of phrases which actually imply an interaction but may not appear to be a simple form of "A is bound to B" sentences. These phrases are actually considered interactions solely from the semantic perspective and do not even contain the frequent key words associated with interaction statements. This leads to a situation where training a system to identify protein interactions based on "symbol statistics" may not work really well and generate false negatives. Second is the observation that some of the key terms in protein interactions such as p53/Creb which actually represents a co-operation in certain excerpts may also be present anywhere in the same abstract without being a part of an interaction sentence. The presence of a large number of such patterns biases PPM to mark all such patterns in the test set increasing the number of false positives. Third, we notice that the terms present in protein interactions are generic English terms that can also be present as a part of interaction phrases that have nothing to do with proteins. These phrases could be talking about interactions amongst two chemical entities that are not really proteins. This increases the false positives as well. The rationale that, by identifying such interaction phrases and then performing post processing that includes only interactions with a protein name does not work well. This is because; more often than not articles contain interaction statements that have pronouns for protein names. By using a rigid model as described above, we would lose valuable information and jeopardize the entire objective of such an automated entity extractor.

5.2 Challenges in Datasets

Some important contributions from this thesis will benefit the research community attempting to perform data mining for entities in biological data. To construct a meaningful, comprehensive, suitably sized training set with a good quality, we curated the Intact data archive. Manual curation of Intact dataset helped us narrow down the paragraphs of evidence text. We have created a data archive of 1169 protein interaction phrases that can be used by other researchers. The complex nature of the protein interaction phrases required considerable domain knowledge for the manual curation. This phase could not be automated as the automation in itself is the objective of our study. Also, a naïve string matching algorithm would fail to identify biologically significant interaction phrases. Hence we created our own datasets. The unavailability of a standard data set for evaluating results also implied a concordant annotation of the test abstracts with the same consistency as performed for the training sets. A change in the consistency would definitely lead to a bias in the results observed. Also the use of domain knowledge aided in analysis of each of the variants being studied and the modifications required for an improved accuracy. Without an in-depth knowledge of what is being studied, improvements would be impossible. Our choice for PERL as a post processing tool comes from our understanding that the post processing involves the extension of strings and PERL's suite of regular expressions would be most powerful in achieving this. Some of the challenges in the datasets for training and our solutions to them had a direct impact on our results.

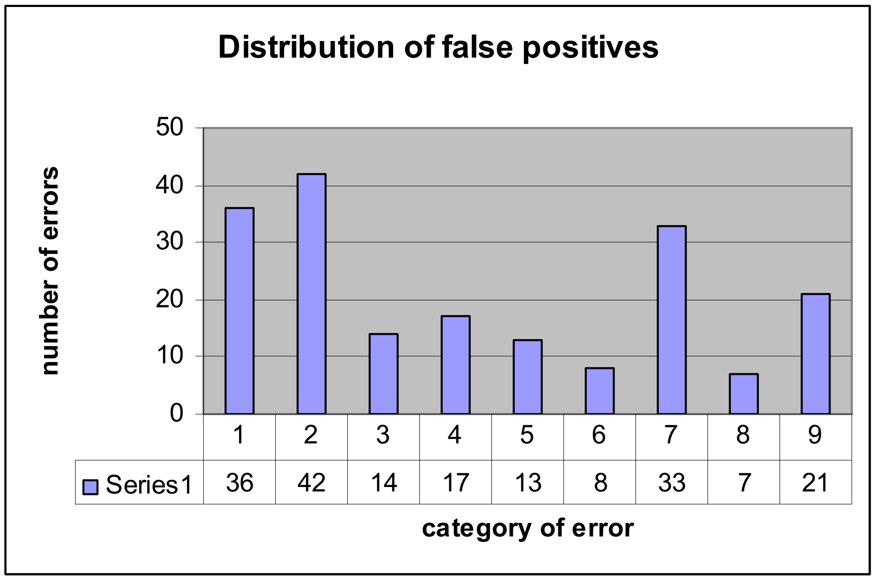
5.3 Analysis of False Positives

The analysis of the kind of false positives observed leads us to a few plausible solutions which could be adopted as a post processing technique. The table (Table 15) gives a complete summary of the kind of errors frequently observed and a suitable solution for the same.

Category of error	Reason	Example	Solution	Count
1	Protein name appears without an interaction phrase	inase defective mutant of Cdc2 fail E2F.3	Check for occurrence of at least 1 interaction term from a list of terms	36
2	Interaction terms appear in the sentence without an actual interaction implied	complex that also shares subunits with SAG kDa complex and the 100 kDa	Check for presence of atleast 1 protein name in the list , but pronouns can also be present and the rule is rigid	42
3	Protein name / entity combinations appear with slashes similar to an interaction phrase	Rag A/Gtr1p are G proteins and lix-loop-helix/periodicity/AhR non-Skp1-Cdc53/Cullin-F-box protein function for the fission yeast	Confirm an interaction keyword in the following or preceeding terms	14
4	DNA/mRNA terms in interactions but not as protein interactions	DNA lesions, interaction with PCNA -MLL/FBP17-3' fusion mRNA Sites of TAZ mRNA and	If DNA/mRNA key terms are present, eliminate the phrase	17

		protein		
Category of error	Reason	Example	Solution	Count
5	Subunit names such as “alpha, beta, etc” close to protein complexes but without any interaction keyword	with 2alpha:1beta:1gamma subunit ciation of alphaII-betaII spectrin may 4alpha and 4beta subunits H chain-IIA colocalize	Subunits should not have DNA/mRNA and should have an interaction term following.	13
6	Protein interaction method of study	study, immunoprecipitation was By using the yeast two-hybrid	Length of the phrase coupled with protein name presence	8
7	Terms representing the domain ends such as “N terminal or C terminal”	terminal domain of RNase E demonstrates that the carboxy-terminal domain of RNase E h han at the COOH terminus of VC	Length of phrase with protein name combinations	33
8	Other chemical entities and interactions with them	Copper – protein interaction HSP1 – lipid interaction	Identify protein plus frequently occurring elements and chemical molecules	7
9	Actual False positives not in the above groups			21

Table 15: Analysis of False Positives



	1	2	3	4	5	6	7	8	9
▢ Series1	36	42	14	17	13	8	33	7	21

Figure 11: Distribution of false positives

For example the observation that the presence of a slash amidst protein/gene names can appear as a protein interaction phrase (Figure 11), one could use a script that detect all such phrases containing the slash and check for the presence of common interaction keywords (a dictionary of terms) in the same phrase. If such a term is not found the phrase could be eliminated from the list of hits. Though some solutions are mentioned in the table there may be a few problems in implementing the entire set of solutions. For example, using a rigid rule such as the presence of a protein name along with an interaction term always could lead to the loss of some of the true positives. Therefore, a trade off must first be arrived at and a suitable solution scheme implemented to reduce the false positives to the minimum must be used. We certainly are very keen to study the results that may be generated by implementing some of our proposed solutions. At this moment we leave this as an open problem to be tackled by the interested.

6. CONCLUSIONS

From our expansive experiments with PPM as a text compression technique for mining key phrases and the results we observe, we can conclude that PPM promises to be a very suitable approach for identifying key phrases of interest from scientific articles. PPM identifies key phrases with excellent precision and recall when provided with a training set of suitable size. This varies from one application to another but by having good models with a high degree of distinguishing ability between the phrase of interest and the rest of the abstract, the results can be improved further. In cases of applications where the key phrase of interest is well defined, as in gene function phrases, PPM shows tremendous accuracy. One may need to arrive at a balance between precision and recall in specific cases of applications where the key phrase may require semantic interpolation. When the identification of all relevant phrases is the focus, a lower precision value may be tolerated. When the presence of false positives causes an unfavorable impact, a lower recall may be used. A good balance is achieved by defining the problem well and constructing comprehensive training sets. PPM, therefore promises to be a vital tool for the automation of text mining in biology.

6.1 Future Work

These are the predominant reasons for the deviation in the F score across the two problems being addressed. Some solutions to these may well be

1. An increased dataset that contains more specific and well curated interaction phrases. As we have used only a portion of the total Intact archive (1600 interaction phrases) we could possibly achieve an increase in the performance by increasing the size of the training set two or three folds.
2. Follow an iterative approach wherein extracted protein interactions are added to the existing training set to create a newer model and then use this model to perform the actual test. This could be studied to identify the optimal number of iterations required before the performance plateaus or begins to drop.
3. Further develop a model class that contains just the interaction terms without protein names to further reduce the false positives that arise.
4. Test the performance of TMT using a token of a higher level say a word based or a sentence based analysis rather than a character based analysis.
5. An analysis of the distribution of errors indicates the need for a post processing scheme to keep the false positives to a minimal level. This could also mean a suitable tradeoff between precision and recall depending on the end application sought.
6. Implementing the solutions for false positives as mentioned in (Table 15).

REFERENCES

Ahmed, S.T., Chidambaram, D., Davulcu, H., Baral, C., 2005. IntEx: A Syntactic Role Driven Protein-Protein Interaction extractor for Bio-Medical Text. In *Proceedings Of workshop ACL-05/ISMB-05* (pp. 54-61).

Bell, T., Cleary, J.G., Witten, I.H., 1990. Text Compression. Prentice Hall.

BioCreAtIvE - Critical Assessment of Information Extraction systems in Biology. *http://biocreative.sourceforge.net/*

Blaschke, C., Hirschman, L., Valencia, A., 2002. Information Extraction in Molecular Biology. In *Briefings In Bioinformatics* (pp. 154--165): Oxford Journals

Celikel, E., 2005.A Cryptographic Approach to Language Identification: PPM .In *the Proceedings of the 7th Int'l Conference on Enterprise Information Systems* (ICEIS).

Chiang, J., Yu, H., 2003. MeKE: Discovering the functions of gene products from biomedical literature via sentence alignment. In *Bioinformatics* (pp. 1417--1422).

Cleary, J., Witten, I., 1984. Data compression using adaptive coding and partial string matching. In *IEEE Transactionson Communications* (pp. 396-402).

Cleary, J.G., Teahan, W.J., Witten, I.H., 1995.Unbounded length contexts for PPM.

Collier, N., Nobata, C., Tsujii, J., 2000. Extracting the names of genes and gene products with a hidden Markov model. In *Proceedings of Conference in Computational Linguistics* (pp. 201--207.)

Fukuda, K., Tsunoda, T., Tamura, A., Takagi, T., 1998. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing '98* (PSB'98) (pp.707—718).

Fundel, K., Guttler, D., Zimmer, R., Apostolakis, J., 2005. A Simple Approach for protein name identification:prospects and limits. In *BMC Bioinformatics, Vol 6, supplement 1.*

Google. *www.google.com/*

Grünwald, P, 1996. A minimum description length approach to grammar inference. In *Symbolic, Connectionist and Statistical Approaches to Learning for Natural Language Processing* (pp. 203--216).

Ehler, F., Ruch, P., 2004. Preliminary report on the biocreative experiment. In *Proceedings of BioCreative Workshop, 2004.*

Hiroyuki, A., Kazuhiro, K., Takashi, I., Shigeichi, H., 2005. A PPM* algorithm using context mixture. In *the Journal of IEIC* (pp. 35--40).

Huffman, D.A. (1952) A method for the construction of minimum redundancy codes. *In Proceedings of IRE* (pp. 1098--1101).

IntAct data archive for text mining. ftp://ftp.ebi.ac.uk/pub/databases/intact/current/various/data-mining

Jelinek, F., 1985. Self-organized Language Modeling forSpeech Recognition. In *IBM Report.*

Kim, J., Park, J., 2004. Annotation of gene products in the literature with Gene Ontology terms using syntactic dependencies. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)* (pp.528--534).

Knight, K., 1999. Mining on-line text. In *Communications of the ACM* (pp. 58-61).

Krallinger, M., Padron, M., Valencia, A., 2005. A sentence sliding window approach to extract protein annotations from biomedical articles. In *BMC Bioinformatics, 6, Suppl.1, S19.*

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

Koike, A., Niwa, Y., Takagi, T., 2004. Automatic extraction of gene/protein biological functions from biomedical text. In *Bioinformatics* (pp. 1227--1236).

Lee, KJ., Hwang, YS., Rim, HC., 2003. Two-Phase Biomedical NE Recognition based on SVMs. In *Proceedings of the ACL 2003 Workshop on NLP in Biomedicine* (pp. 33–-40).

Lelewer, D., and Hirschberg, D. (1987) Data Compression. In *ACM Computing Surveys* (pp. 261--296).

LLL'05 contest. Learning Language in Logic Workshop. http://genome.jouy.inra.fr/texte/LLLchallenge/

Moffat, A., 1990. Implementing the PPM data compression scheme. *In IEEE Transactions on Communications*(pp. 1917--1921).

National Center For Biotechnology Information. *www.ncbi.nlm.nih.gov/*

Nelson, M., 1991. Arithmetic Encoding and Statistical Modeling is data compression. In *Dr.Dobb's Journal*.

Otasek, D.,Brown, K., Jurisica, I., 2006. Confirming protein-protein interactions by textmining. In *proceedings of SIAM Conference on Text Mining*.

Ratnaparkhi, A., Roukos, S., and Ward, R. T., 1994. A Maximum Entropy Model For Parsing. In *Proceedings of the International Conference on Spoken Language Processing* (pp. 803--806).

Raychaudhuri, S., Chang, JT., Sutphin, PD., Altman, RB., 2002. Associating gene with Gene Ontology codes using a maximum entropy analysis of biomedical literature. In *Genome Research* (pp.203--214).

Rooij, S.D., 2003. Methods of statistical data compression. *Master's thesis. University of Amsterdam.*

Seki, K., Mostafa, J., 2003. Toward Database Curation Support in Biology: Automated Gene Function Identification from Texts. In *TREC Genomics Track. # lair.indiana.edu/research/capris/papers/ lair04-02.pdf*

Shannon, C.E., 1948. A Mathematical Theory of Communication. In *Bell System Technical Journal* (pp.379-423).

Sommer , R.L., Guimaraes, A.P., 2004. Applying the zipping method to Barkhausen noise in order to estimate the degree of (dis)order. In *Journal of Magnetism and Magnetic Materials* (pp 551--552).

Stoica, E., and Hearst, M. Predicting Gene Functions from Text Using a Cross-Species Approach. In *Proceedings of 2006 Pacific Biocomputing Symposium PSB*.

Tanabe, L., and Wilbur, W.J. Tagging Gene and Protein names in Full Text Biomedical Articles.*In Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain (ACL-2002)* (pp. 9--13).

Teahan, W.J., 1998. Modeling English Text. *Ph.D. thesis, Dept. of Computer Science, The University of Waikato.*

Teahan, W. J., Harper, D. J., 2001. Combining PPM models using a text mining approach. In *proceedings of IEEE Data Compression Conference*.

Teahan, W.J. Power point presentation on text mining toolkit.

TREC - Text Retrieval Conference. *http://trec.nist.gov/*

Viterbi, A.J., 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm.In *IEEE Transactions on Information Theory* (pp. 260--269).

Wikipedia. www.wikipedia.org

Witten, I.H., Bray,Z., Mahoui,M., Teahan, W.J., 1999. Using language models for generic entity extraction. In *Proceedings of ICML Workshop on Text Mining*.

Witten, I. H., Neal, R., Cleary, J.G., 1987. Arithmetic Coding for Data Compression. In *proceedings of Communication ACM* (pp. 520—540).

Yeates, S., Bainbridge, D., Witten, I.H., 2000 . Using compression to identify acronyms in text. In Storer, James A. and Martin Cohn (eds). In *proceedings of Data compression Conference* (pp. 582).

Yeates, S., Witten, I.H., 2000. On tag insertion and its complexity.In *Proceedings of PRICAI'2000 Workshop on Text and Data Mining* (pp. 52--63).

Ziv, J., Lempel, A., 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory* (pp. 337–343).

APPENDICES

VITA

Arvind Kumar Thirumalaiswamy Sekhar

sai_arvin@yahoo.co.in
(317) 525-9648
725 W Walnut Street Apt G
Indianapolis IN 46202

Education

Master of Science, Bioinformatics *Feb 2008*
School of Informatics, Indiana University GPA 4.00
Key courses: Machine learning and Pattern recognition,
Data mining, Bioinformatics, XML, Structural bioinformatics

Bachelor of Technology, Bioinformatics (*Honors*) *May 2006*
Vellore Institute of Technology, India GPA 3.92
Key courses: Object Oriented Programming, Algorithms,
Machine Learning, Molecular biology, Cell biology, Immunology, Genomics

Professional Experience

Bioinformatics Graduate Co-op *May2007-Aug2007*
IBM Research, T.J.Watson Center,NY
- Performed parallel programming for proteomics analysis and biomarker discovery.
- Implementation on an opteron cluster using Message Passing Interface and visual C++.
- Developed code for meshing LCMS spectra and peak identification.

Informatics Programmer Intern *Jan 2007-May 2007*
DowAgroSciences,Indianapolis,IN *Sep 2007-Dec 2007*
- Implemented PERL scripts for Expressed Sequence Tag assembly.
- Workflow design, development and testing.(Pipeline Pilot).
- Transfer of MySQL to new cluster and execution of assembly on the cluster.
- Design and development of a relational database for capturing EST data.
- Development of GUI/Web interface for the application.

Programmer Intern, Bioinformatics *Oct 2006-Jan 2007*
University Information Technology Services, IN
- Implemented data format conversion tools for data residing within the FASD.
- Deployed Perl DBI for data retrieval and analysis and constructed web interface using CGI.
- Integrated Gene Database into CLSD at UITS using DB2.

Research Assistant *Oct 2006-Jan 2008*

Indiana University, School of Informatics, IN

- Used Text encoding and entropy detection for mining gene functions from Scientific abstracts present in PubMed.
- Utilized Prediction by Partial Matching (PPM) through Hidden Markov Models.

Skills

- **Languages:** C, C++/visual C++, C#, PERL, SQL, JAVA (Beginner), and XML.
- **Databases:** Oracle 8i/9i/10g, MySQL, SQL.
- **Web Designing Tools:**CGI**,** HTML,Dreamweaver,FrontPage.
- **Platforms:** Windows XP/2000/NT, DOS, UNIX, Linux, AIX (working knowledge).
- **Packages:** MatLab, MS Office, AutoDock, Swiss PDB viewer, Argus LAB, Glimmer, ESTScan, PHRAP, CVS.
- **Workflow software:** Pipeline pilot.

Honors

- Regional Finalist, Microsoft Imagine Cup Software Development 2007.
- Paper on "Workflow based framework for life sciences" accepted by the Computational Biology and Chemistry journal.
- Poster on High resolution analysis of LCMS data presented at IBM, TJ WATSON research center.
- Poster on EST assembly accepted for the Indy '07 Bioinformatics Conference.
- University Fellowship Recipient, Indiana University.
- Cleared the Japanese Language Proficiency Test conducted and certified by the Government of Japan.
- Structure of Sphingosine-1-phosphate receptor deposited with the Protein Data Bank.
- Silver and Bronze Medal holder of Duke of Edinburgh Award for students.
- MERIT certificate from Association of Mathematics Teachers of India, New South Wales talent exam (English and Mathematics), Geography talent test.