A Supervised Machine Learning Approach of Extracting and Ranking Published Papers Describing Coexpression Relationships among Genes

Richa Tiwari , Chengcui Zhang, Thamar Solorio, Wei-Bang Chen,

University of Alabama at Birmingham
Department of Computer and Information Sciences,
115A Campbell Hall, 1300 University Boulevard,
Birmingham, Alabama 35294, Phone: 205.934.2213, Fax: 205.934.5473
{rtiwari, zhang, solorio, wbc0522}@cis.uab.edu

Abstract. In this chapter, we describe a framework to extract information about coexpression relationships among genes from published literature using a supervised machine learning approach, and later rank those papers to provide users with a complete specialized information retrieval system. We use Dynamic Conditional Random Fields (DCRFs), for training our classification model. Our approach is based on semantic analysis of text to classify the predicates describing coexpression rather than detecting the presence of keywords. Our framework outperformed the baseline by almost 52%, with DCRFs showing superior performance to Bayes Net, SVM, and Naïve Bayes classification algorithm. In our second experiment, the comparison of our ranked results to that of PubMed and Google demonstrates that our proposed model performs better than both in distinguishing a positive paper from a negative paper. In conclusion, this chapter describes a specialized classification and ranking framework that can retrieve articles that discuss coexpression among genes.

Keywords: Relationship extraction, Natural Language Processing, Dynamic Conditional Random Fields, Information extraction, Information Retrieval.

1 Introduction

With the advent of technology and Internet, information exchange and storage has increased tremendously. There is a huge amount of information available in almost all fields on the Internet. The technological development has led biomedical research scientists to publish and share their findings and results online. PubMed is one such source where people can find a large amount of data and publications [1]. According to a recent fact sheet produced by MEDLINE a component of PubMed, it consists of almost 712,615 indexed citations by the end of 2009 [2]. Figure 1 shows some of the statistics of the trend of citations in the past five years in MEDLINE. We can clearly see that there is 6% increase in the indexed citations from 2008 to 2009 in MEDLINE/PubMed alone. Such a vast source of information being readily available is a great resource for new research and hypotheses. However, this also leads to some

of the problems related with handling massive amount of data, such as data management, storage, extraction of precise information and appropriate information retrieval.

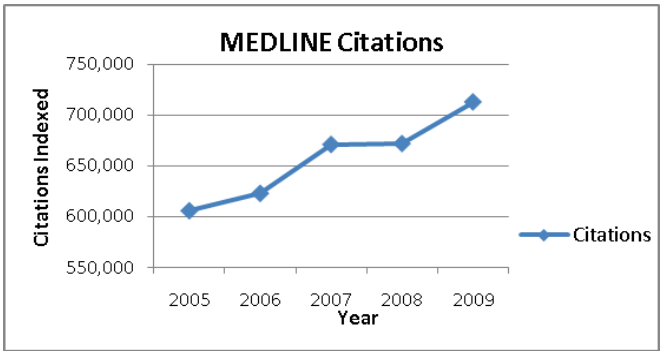

Fig. 1. Graph showing the increase in the rate of citations being indexed in MEDLINE in past 5 years.

It is often times essential to obtain very precise and relevant information from published papers without wasting too much time in reading the whole paper. As research is progressively increasing in the biomedical discipline more and more papers are being published. Information extraction from these papers has become a very challenging task for both computer scientists and computational biologists. A computer scientist can help biomedical researchers in managing this data by building semantic information extraction and retrieval tools that can suit their needs.

One piece of such precise information that is needed by biomedical researchers is Gene-Gene Relationships, often referred to as Protein-Protein Interactions (PPI). These relationships can lead to the discovery of new hypotheses. Several kinds of interactions exist between and among genes and one such relationship is called "Co-expression" relationship. If two genes are expressed together, they are said to be coexpressed. This is a very important relationship and property among genes. If an unknown gene is expressed together with a known gene, we can easily assume that there exists some functional relationship between them. And we can then determine some of the properties of this unknown gene based on the known gene. These coexpressed genes can share the same pathway and can lead to several other functionalities. Also, the extracted relationships can be further used to build up networks that can describe complex biological pathways.

Often this coexpression relationship can be determined by various biological experiments such as microarray experiments and immune staining. There are publicly available gene expression repositories based on the results of microarray experiments that can give ranked lists of coexpressed genes [3]. The papers that are published with the results of these microarray experiments are very rich in information and hold a lot of other useful details related to the experiments that were performed and the by-products of this coexpression. Abstracts of these papers are readily available on PubMed, but they do not have extended information as what a full length papers can provide us with. Full length papers can help others to replicate those experiments, make more hypothesis based on the information present in the paper, etc. Often, just

information extraction from the paper is not enough and we need to present this information in a ranked list. This ranked list of papers or documents can help the user to select the critical information from the top of the list more easily. For example, when we perform a search in an information retrieval system, we assume that the top few documents/web pages are the most relevant to our query. Similarly it is also essential to provide researchers with the papers that have the coexpression relationships as the main content of the paper. To provide the researchers a ranked list of papers based on the information content, it is essential to have a good extraction model first. Once we have a good information extraction model, we can provide a better retrieval results based on it.

To the best of our knowledge, there is no existing tool that can extract predicates that talk about coexpression relationships between and among genes from published articles based on text analysis and information extraction techniques. A work presented by Tiwari et al. is the first step towards extracting such relationship from published literature [4]. Our goal in this chapter is to present a framework that can retrieve papers that talk about gene-gene coexpression relationships, by using machine learning approach to first extract the predicates describing these relationships and then using our scoring scheme to rank the retrieved positive papers. Good retrieval results can be achieved based on a good extraction model in the back-end of it and hence we propose a retrieval framework that is based on our sentence classification model. As our classification model is especially trained to extract coexpression sentences irrespective of whichever ways they have been expressed in the paper, our retrieval system which is based on it will be able to rank the papers better even if the query terms use only one of the possible ways to this concept. We have used Dynamic Conditional Random Fields (DCRFs), a graphical probabilistic model, trained on these predicates [5]. Afterwards, we present our results by testing this model on full-length biomedical papers collected from PubMed Central, a free digital archive of biomedical and life sciences journal literature of U.S. National Institute of Health (NIH). For the information retrieval part, we have created our own scoring scheme to rank the papers for 5 different query genes and their coexpression. We then compared our ranking with Google and PubMed for the same set of papers and query terms using a well known rank comparison metric, Mean Average Precision (MAP) and a modified version of MAP that we created to agree with our needs [6]. The ranking comparison results show that our model performed much better than both of these search engines in retrieving the papers that talk about coexpression of genes.

The experiments performed in this work can be divided into two main parts. The first part includes the information extraction part, where we train a DCRFs model that can classify sentences as positive or negative based on whether they talk about gene-gene coexpression or not. In the second part we present the evaluation results of the proposed information retrieval system that ranks a paper according its relevance to the query gene.

The rest of the chapter is arranged as follows. Section 2 discusses the related work in this field. Section 3 introduces the methodology of our framework, and we present and analyze our results in Sections 4 and 5.

2 Related Work

Information extraction from text is the process of automatically extracting structured data from unstructured data, usually text written in a natural language like English. There has been a lot of work done in the area of information extraction in the field of biomedical sciences. Noticeable conferences such as, Knowledge Discovery and Data Mining (KDD) challenge held in 2002 involved extraction of information from full-length papers in the field of biomedical research [7]. Another major conference in this field is BioNLP, which encourages research in natural language processing of biological text.

Aaron M. Cohen and William R. Hersh give an up-to-date survey of work done in biomedical text mining field [8]. Some effort in this area incorporates extraction of information such as named-entities, different medical terms, relationships etc from either the abstracts of the published literature or the full-length papers themselves. In the past, researchers have used various natural language processing techniques such as hand-written rules based on linguistic knowledge, to extract information from text [9]. However, these approaches are very expensive and require a lot of time and effort from domain experts, which involves writing all the rules manually. Hence, more statistical approaches such as machine learning have been introduced to aid the Information Extraction task. One of the early works done in this field was using Naïve Bayes probabilistic model to extract information about protein localization pattern from MEDLINE abstracts [10]. Similarly, Hidden Markov Models, Neural Networks and similar stochastic classification algorithms have been successfully used in extracting information from text [11]. Few other works towards extracting interactions between biomedical entities includes use of machine learning classification algorithms like Support Vector Machine (SVM) and Neural Nets. Probabilistic graphical models like Hidden Markov models (HMM) and Conditional Random Fields (CRFs) have proven to be quite successful in classifying and extracting relationships between biomedical entities [12, 13]. CRFs are undirected graphical models developed by Lafferty et al. in 2001, and have been applied to various text mining tasks such as table extraction and named entity extraction in biomedical text [14, 15]. Bundschus et al. used linear chain CRFs to extract gene-disease relationships from biomedical literature [16]. DCRFs are an extension of CRFs and have been shown to outperform CRFs in natural language chunking tasks [5].

Gene-gene coexpression relationships extraction is a subtask of relationship extraction, but differs in the way that this relationship is expressed in several ways in natural language in actual text. This is a widely researched area in computational biology and both manual and automatic work is being done in this area. There are existing databases that contain manually curated 12,000 Medline articles for protein-protein interactions (PPI) [17]. Work done towards automatic extraction of PPI using natural language processing tools (NLP) includes Miyao et al. in which they evaluated the contribution of natural language parser to protein-protein interactions (PPI) [18]. They showed that the appropriate natural language parsers can help researchers extract information such as PPI from unstructured data such as published literature. Another work in this direction proposes to use a rich NLP feature vector set and

Support Vector Machine classification algorithm to extract PPI from sentences [19]. Another approach towards extracting PPI from papers is the co-occurrence counts of words, that is, if the two entities occur often or have a textual pattern of occurrence in a paper, then we can predict if a relationship among them exists or not by using statistical measures like pointwise mutual information, chi square, etc. A work by Mooney et al. combines this co-occurrence statistics of words approach as well as the information extraction approaches that use classification algorithms to extract sentences and presents an IR system for PPI from Medline [20]. SUISEKI, is another framework developed for detecting PPI from published text by Valencia et al. in 2002 [21]. Generally there is a pattern of expressing a relationship between two entities, for instance the relationship word, usually a verb, will occur in between the entities in the text and hence can be extracted if we know the entity boundaries. This tool uses a set of predefined rules or frames along with the other statistical and linguistic tools to extract PPI from literature. The rule based extraction makes the process non expandable and time consuming to prepare. It is not manageable to collect all the possible rules for sentences and thus the retrieval system will only be able to retrieve limited number of documents. Coexpression relationships are not necessarily written in this format, certain challenges can occur in classifying those using fixed patterns. For example, there are several different words that can express this interaction among genes. Some of the commonly used words are – "Coexpress", "expressed together", "both were expressed", "up-regulated", etc.

In this approach we have provided a complete framework for extracting and retrieving papers about coexpression relationship among genes. But, as our classification and indexing scheme is independent of any specific retrieval algorithm it can be used with any technique. Also, as our work focuses on the retrieval of papers based on not just the query keywords but semantic meaning of those keywords, it provides a better result than PubMed and Google in the case of papers talking about coexpression relationship among genes. This can be seen from the comparison results later in Section 4 of this chapter.

3 The Proposed Approach

Automatic relationship extraction from unstructured machine-readable text is a complex and challenging task. Following recent successes in using CRFs for relationship extraction, we have explored their use in our application. Relationship extraction is comprised of annotating the unstructured text with the entities involved and the relationship between those entities. In our case, entities are the genes and the relationship terms are the ones that express coexpression relationships. As illustrated in Figure 2, our framework for the sentence classification task can be divided into five steps: 1) Data collection, 2) Pre-processing, 3) Feature extraction and class assignment, 4) DCRFs classification model training, and 5) Feature and class analysis.

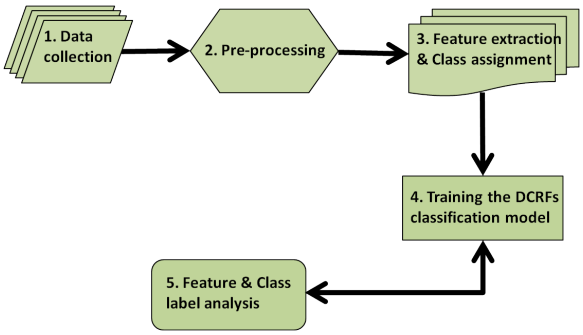

Fig. 2. Block diagram of our proposed sentence classification framework.

We also prepared a scoring scheme and performed rank comparison with Google and PubMed. Figure 3 represents our information retrieval and rank comparison model in detail.

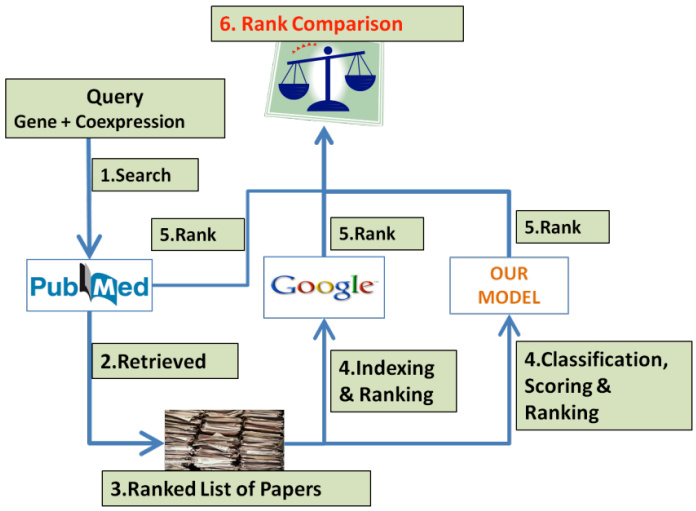

Fig. 3. Block diagram showing our paper ranking and comparison scheme.

3.1 Data Collection

For the sentence classification purposes, we collected 500 full-length papers from PubMed and manually divided them into positive (285) and negative papers (215) depending upon whether a paper contains predicates about some genes being coexpressed and not just the coexpression process in general. A paper that contains positive sentences has both gene names as well the terms defining the coexpression relationships in them. This collection helped us to prepare a model that was later used

for testing the papers in our retrieval process. Figure 4 gives some example sentences that help understand the difference between the positive sentences (sentences that talk about gene-gene coexpression) and negative sentences (sentences that talk about just coexpression in general or do not talk about coexpression at all). Overall, we have 15010 sentences in our repository to experiment with.

For the second part of our experiments, we collected a different set of papers from PubMed using five different query genes. The five query genes were – "Bcl-2 coexpression", "ErbB coexpression", "IL coexpression", "Myc coexpression", and "p53 coexpression". In total we collected 500 papers by taking the top 100 papers for each query. These papers were tested against the DCRFs model created in the first part, scored and ranked.

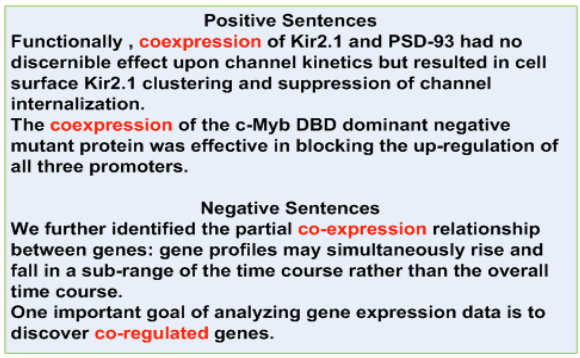

Fig. 4. Examples of positive and negative sentences in literature.

3.2 Pre-processing

The collected papers are all in PDF format and were converted into text using the PDFbox java library [22]. Once these papers were converted into text, we deleted the sentences or parts from the papers that we believe do not provide any useful information about coexpression relationships. These parts include – everything before the abstract section of the paper and everything after the acknowledgment section of the paper. As mentioned earlier, relationship extraction includes tagging the entities and the relationship among them. In our case these entities are gene names. Hence, we assume that the sentences that do not contain the desired entities (gene names) usually would not correspond to any useful information about the relationship. We consequently deleted all the sentences that do not contain gene names. To determine if any gene name is mentioned in a given sentence, we tagged each word in a sentence using the GENIA tagger, a freely available tool that has been trained using HMMs to identify gene names in any given text [23]. The sentences that are not tagged by the GENIA tagger as containing gene names are discarded from the succeeding steps.

3.3 Feature Extraction and Class Assignment

This step is performed for the task of building a classification model. Feature extraction is a very important task in machine learning techniques. It is important to provide a good set of features to any machine learning algorithm, if we want the model to perform well. However, there is no fixed set of features that can be used to improve the performance of any model, as they are very task specific.

The basic set of features is extracted using the GENIA tagger. GENIA parses the sentences and tokenizes them into words. It then assigns features to each word. These features include, the root form of the word, e.g., the word *coexpression* has the root *coexpress*. It also performs grammatical analysis of each sentence and extracts the part of speech tag feature for each word. Part of speech (POS) feature of a word describes the grammatical role of the word in the sentence, e.g., the word "books" has a POS tag as *noun* in the sentence "Books are made of paper ink and glue". But the same word is tagged as *verb* for the sentence "book your flight soon". Another set of features that GENIA tagger extracts is often referred to as the *Chunk tags*. Chunk tags are the tags for the constituent phrases in a sentence. A constituent phrase is a group of words that functions as a single unit in a sentence. Therefore, all the words in a noun phrase will get the chunk tag NP (Noun Phrase) as a feature. It is worth noting that a biomedical entity term will be assigned a chunk tag as 'B-NP' by the GENIA tagger. Finally as mentioned earlier, GENIA tagger also assigns biomedical entity tags to the entities, such as "B-protein" and "B-RNA" and these tags act as additional features of those words. The final basic feature set can be divided into two categories:

Local:

- The word (w_i , where w indicates the word and i indicates the position of the word) itself and the root of the word
- POS tag of the word and its chunk tag, i.e., the tag at the phrase level

- Biomedical named-entity tag

Contextual:

- POS and chunk tags of the words w_{i-3}, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, w_{i+3}
- Biomedically named entity tags of the words w_{i-3}, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, w_{i+3}

We train a DCRFs model on the class labels assigned to words in a way as follows:

- RE class labels assigned to the words that express the coexpression relationships
- GE class labels assigned to the gene name words that are involved in the coexpression
- Chunk tags assigned to the rest of the words in the sentence

Overall, we have 12 class labels including the GE, RE and all the chunk tags, out of which we are interested in classification of RE and GE labeled words by our model. These are the basic or the original set of features and classes that we started our experiment with, and later on we refined the feature set by analyzing the results in Steps 4 and 5. Table 1, shows an example of labeled sentence with each word labeled according to their phrase tag/chunk tags and GE and RE tags.

Table 1. Example of class labeled words in a sentence

Words	**Class Tags**	**Meaning**
We	B-NP	Noun Phrase
conclude	B-VP	Verb Phrase
that	B-SBAR	Subordinating Conjunction
coexpression	**RE**	**Relationship Word**
of	B-PP	Propositional Phrase
LacZ	**GE**	**Gene Name**
and	B-NP	Noun Phrase
M71	**GE**	**Gene Name**
occur	B-VP	Verb Phrase
frequently	B-ADVP	Adverb Phrase
when	B-ADVP	Adverb Phrase
the	B-NP	Noun Phrase
two	B-NP	Noun Phrase
gene	B-NP	Noun Phrase
be	B-VP	Verb Phrase
present	B-ADJP	Adjective Phrase
in	B-PP	Propositional Phrase
cis	B-NP	Noun Phrase

3.4 Training the DCRFs Model

We perform supervised learning using DCRFs, a probabilistic graphical model that can be effectively used for sequence classification problems. It is a generative model that relaxes the Markov assumption of HMMs regarding the input and output sequence. DCRFs combine the concept of CRFs (conditional probability distribution which allows rich feature set) and Dynamic Bayesian Network (DBN). DCRFs calculate the conditional probability distribution $p(y/x)$ of output label sequences y given a particular observation sequence x, rather than finding the joint probabilities of both label and observation sequence [12]. Equation 1 shows how to calculate conditional probabilities of output labels given the input observations using DCRFs.

$$p(y/x) = \frac{1}{z(x)} \prod_{t} \prod_{c \in C} \exp\left(\sum_{k} \lambda_k f_k(y_{t,c}, x, t) \right). \quad (1)$$

where $f_k(y_{(t,c)},x,t)$is referred to as the transition feature function which is equivalent to transition probabilities in HMM. λ_k is the learned parameter vector in the model. $Z(x)$ represents the normalization function, and K is the number of feature functions. t is the time step (state) and C being the set of clique indices. A clique in an undirected graph G is a set of vertices V such that for every two vertices in V, there is an edge connecting them.

In our framework we have used a GRaphical Model for Mallet toolkit (GRMM) which has a java implementation of DCRFs [24]. Once we have the training data ready, we run Mallet for DCRFs algorithm and train a classification model on the classes present in the training set. This model learns the conditional probability values for the occurrence of each class given any input sequence. Subsequently when a testing data is given to it, the model will assign class labels to each of its input words based on the learned parameters.

Once the input files are tagged with their true class labels and features, we divide the data into training and testing sets. We also create a small subset of tagged data referred to as the development set, which consists of 4,109 sentences (1,333 positives ones and 2,776 negative ones). A development set is a set of testing data that is used for parameter tuning and feature assessment and is different from the main testing file. This set is prepared separately from the main testing set such that while parameter tuning our model does not get biased towards our main testing file. This development set was used as the testing data set in the next step to select the appropriate set of features and class labels. In other words, the next step is to determine which features to use for predicting the class labels.

3.5 Feature and Class Selection

In this section we present our process of feature selection and class label analysis. We use our development set as the testing data for this process and performed several experiments with different feature combinations. DCRFs are good for representing complex interactions between class labels. In our framework, this will refer to the interaction between class labels for genes that are actually involved in coexpression

relationships and the labels for coexpression relationship terms. This leads us to experiment with various sets of appropriate class labels because gene name extraction in itself is a big area of research and we are more interested in finding the coexpression terms. Finally we come up with a set of features and class labels that have the strongest correlation with the classification variable, i.e., the feature and class label combination that gives the best results for tagging RE labels. We experimented with different combinations of features and classes, shown in Figure 5. Each of these experiments and their results are explained in more details in Section 4. The list of experiments performed includes:

1. Experiment with the original feature set, which includes all the features mentioned in Section 3.3.
2. Experiment with trigger words gazetteer with synonym features.
3. Experiment with removing stop words after assigning contextual features to the data set.
4. Experiment with learning different sets of class labels, which include – 1) no GE class labels, such that all the gene names get their chunk tags as their class labels, and 2) all the biomedical entity words get the biomedical entity (BME) tags assigned by the GENIA tagger as their class labels. So, no GE class labels but three other class labels – B-protein, B-DNA, and B-RNA.
5. Experiment with different combinations of settings from Experiments 3 and 4.

Experiment with just a few class labels including B-protein, B-DNA, B-RNA, RE and NA, where NA class label was given to all of the irrelevant words in the sentence (the ones that were given their chunk tags as their class labels in earlier experiments).

Once we decide on an appropriate set of features and class labels, we perform training and testing on our larger data set. The comparison results are presented in the next section.

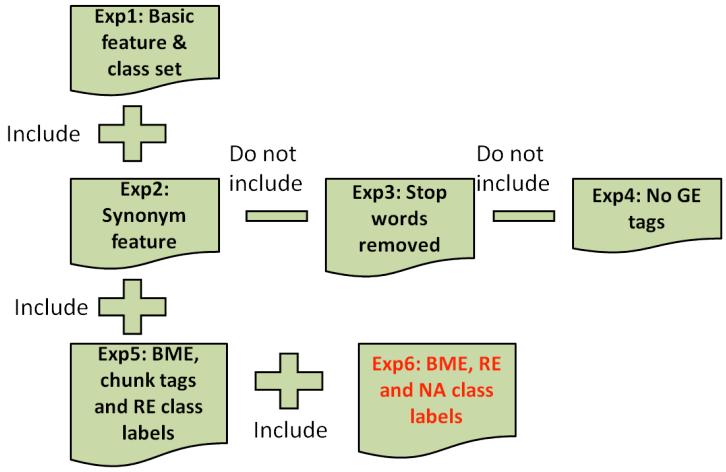

Fig. 5. The process of experimenting with different combinations of features and class labels.

3.6 Scoring Scheme

Once we have a DCRFs model trained to classify sentences talking about coexpression, we use that model to test the second set of 500 papers that we collected using 5 query genes from PubMed. The end product of this testing are the class labels generated for each sentence in the testing papers, which distinguish between a positive sentence and a negative sentence. We propose a scoring scheme to score these test papers in this section. We score a paper based on three main criteria: 1) Number of positive sentences tagged by our model in the paper, 2) Location of those sentences, and 3) Number of query genes tagged by our model in the paper. We implemented a location based weighting of the sentences in the paper, with the *Abstract*, *Results and Discussion* sections of the paper weighted twice as much as the rest of the paper. We made this assumption because often the important results are stated in the *Abstract* section, and it is almost always the case that the *Results* and *Discussion* section contains the result and important observations of the paper. Therefore, if a coexpression sentence occurs in one of these sections of a paper, we can assume that this paper is talking about coexpression of a gene. This zone weighted scoring scheme can be explained by Equation 2.

$$ws = \sum_{i=1}^{4} \left\{ count_s(i) \times w(i) \right\}. \quad (2)$$

In this equation, *ws* is the total weighted score of a paper, *i* is the location or zone and we have 4 different zones. $count_s(i)$ is the total number of sentences at location *i*. $w(i)$ is the weight of each particular zone *i,* which can be either 1 or 2 depending upon whether the sentence was found at *Abstract* and *Result & Discussion* sections ($w(i)=2$) or *Introduction* and *Material & Method* sections ($w(i)=1$).

To calculate the number of query genes tagged by our model, we first prepare a manually built dictionary of all the gene synonyms and the family names of the query genes from PubMed and then perform a dictionary matching from the tagged list of genes by our model to count the number of occurrences of query genes in the whole paper.

Once we collect all three counts, i.e., the number of positive sentences, their location based weights, and the number of occurrences of query genes, we then normalize the whole score by the number of pages in the paper. Hence, the final score assigned to each paper can be calculated as in Equation 3.

$$Score = \frac{tcount + ws + gs}{pages}. \quad (3)$$

Where score indicates the total score assigned to a paper which is the sum of the number of positive sentences (*tcount*), the zone weighted score (*ws*), and (*gs*) the total

number of query gene occurrences, and this sum divided by the total number of *pages* in the paper. This division helps us to normalize the scores relative to the length of the paper. After we have the score of each paper, we rank them with the highest scoring paper getting the top rank.

4 Results

In the following two subsections, we present results of our sentence classification experiment and information retrieval experiments, respectively.

4.1 Sentence Classification

In this section we present the experimental results of our model in detecting the coexpression predicates from text using DCRFs. We started with 500 full length papers in our experiment and performed pre-filtering on it. After down sampling we were left with approximately 15,010 sentences all together. As we used a supervised learning approach, each word in each of these sentences was manually tagged as the ground truth for learning. Out of the total 15,010 sentences we have around 3,130 positive sentences which are only 20% of the total data set. Once the sentences are tagged appropriately, we divide the whole data set into training and testing files and perform training and testing on them using DCRFs. We present our results in the form of Precision, Recall and F-measure. F-measure is the weighted harmonic mean of the both Precision (P) and Recall (R). Figure 6 shows a comparison between the results of our **baseline model** and that of model trained with the **original set of features**. In the baseline experiment we perform the basic term matching from training to testing file. All the terms in the training files that correspond to coexpression terms were collected in a training corpus, and then each testing file was matched against it. If any word in the testing file matches some word in the training corpus, then the file is tagged as positive. This can also be considered as dictionary matching where the dictionary is built each time from a set of training files. The experiment with the original set of features involves experimenting with the set of features mentioned in Section 3.3.

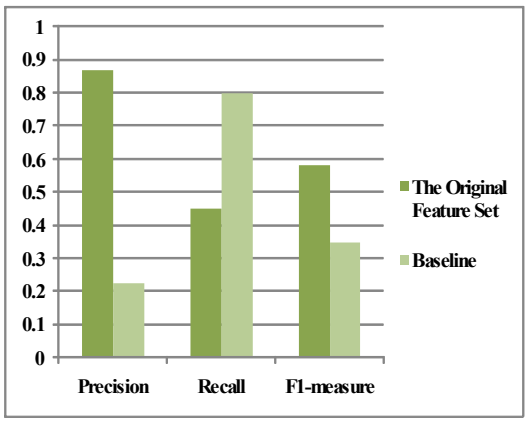

Fig. 6. Comparison of the results obtained by the baseline method and that of the model trained with the original feature set.

We can see that although the baseline approach has a higher Recall score, it has a very poor Precision. This is because the baseline approach is solely based on keyword matching, and the contextual information is missing. One common example is the word "express" along with some other word like "together" or "both". This word, if used alone without a helping word like "both" or "together," does not indicate anything about coexpression relationships, but in the baseline approach every "expression" word will be tagged as positive in a test file. This will result in lots of false positives and lead to low Precision. Our model, trained with just the original set of features without feature selection, is almost 23% better than the baseline model in terms of F-measure.

As mentioned in Section 3.5, we performed some experiments to come up with the best set of features and class labels. Table 2 shows the 10 fold cross validation result of each of those 6 experiments, which include different combinations of features and classes.

The first experiment involves experimenting with **Trigger words** and the results are presented in the second row of the table. Trigger words are the words that most commonly describe the variable that we want to classify. A gazetteer of these words can be created and a model can be tuned on them. We created a list of approximately 200 trigger words by including words like "coexpress" and their inflected forms. Whenever a word in a file matches some word in the trigger word gazetteer, it is assigned a feature referred to as "synonym feature," i.e., tagged with the word "coexpress". This improved the Recall of our system by 6%, which means that our model was able to identify more of the positive sentences. Since we achieved a significant improvement in the performance by adding this feature (as shown in Table 1), we included it for the rest of our experiments.

Table 2. Experimental results with different combinations of features and class labels

	Precision	Recall	F1
Trigger words	0.81	0.51	0.62
No stop words	0.81	0.49	0.61
No GE labels	0.81	0.58	0.67
BME tags as labels	0.82	0.56	0.66
No stop words & BME tags	0.80	0.50	0.61
BME,RE and NA class labels	**0.81**	**0.63**	**0.71**

Stop words are the commonly occurring words like articles and prepositions that do not contain any useful information regarding the semantic content of the text. It is a common practice in natural language processing to filter out these words from the text as a preprocessing step. In the second trial we removed all the stop words from the training files but kept the contextual features for each remaining word. The result of this **No stop words** trial is presented in row 3 of Table 1. Although this approach performed better than the original feature set experiment, we did not achieve any significant improvement in this experiment (as shown in Table 1). This is probably because once these words were removed we also lost their syntactic tags/labels. And as we know DCRFs also learn the relationship between labels, thus losing those labels within a sentence may throw off the model and lead to poorer performance.

As aforementioned, DCRFs help in capturing complex relationship between labels and are useful in chunking task. In our task of relationship extraction, we want to identify the words describing coexpression as well as genes, and the performance of our DCRFs model is influenced by the prediction accuracy of GE as well as RE tags in a sentence. Therefore, we decided to further experiment with the class labels too. In the third experiment, mentioned as **No GE labels** in Table 1, we replaced all the GE class labels for gene names with their chunk tags 'B-NP' assigned by GENIA tagger. In this case we made no distinction between the gene names and the other regular words in terms of the way their class labels are assigned. This leads to the least guessing work in model training and the prediction of class labels for non RE terms as there is a near-deterministic relationship between the input feature and the class label. This approach shows the highest improvement on our dataset, a 9% increase from the experiment with the original set of features, but does not give any specific

information about the genes that are coexpressed. Consequently we did not include this in our final experiment.

Thinking along the same lines as before, we replaced all the GE class labels from the gene name words with the biomedical named-entity tags (not the same as chunk tags) that GENIA tagger assigns to those words. We also assigned those class labels to all the words that are tagged as biomedical entities by GENIA. Hence not just the genes involved in the coexpression have those labels, but all of the gene names, protein names, and RNAs are assigned those labels. Though, our model now had three additional class labels to learn (B-protein, B-DNA and B-RNA), it still performed almost as well as the **No GE labels** experiment (see Table 1, row 5 **BME tags as labels**). However, with this approach, we not only get the coexpression relationship words extracted but also the genes that are involved in the relationships. Recall of our system is also significantly higher, almost 11% higher than that of the experiment with the original set of features.

In the 5th experiment, we combined experiments 2 and 4, i.e., all the biomedical entities got their GENIA assigned tags as the class labels and all the stop words were removed from the sentences. However, this combination did not obtain any performance gain probably for the same reasons as stated earlier for the second experiment.

In the last experiment, we decreased the number of classes from 12 to only 5 by assigning just one class label to all the words that are not useful for us and were given their chunk tags as the class labels. We kept the three biomedical entity name tags as the class labels (B-protein, B-DNA, and B-RNA), the RE class labels and all the other words got an NA class label. We see a high improvement in our Recall (7%) and F-measure (10%) values from experiment 4 above. This was due to the fact that in this case our model had fewer classes to learn and more instances. The noise produced by too many class labels in the form of chunk tags was reduced in this experiment.

Finally by analyzing all of these experiments on the development test set and training set, we came up with the following set of features and class labels for training our model. These include:

1) Local and contextual features as mentioned in Section 3.3.
2) Synonym features as used in experiment 1.
3) Class label RE for words that express coexpression relationships.
4) Class labels B-protein, B-DNA and B-RNA for tagging biomedical entities.
5) Class labels NA for all the other words in the sentence.

Although the Precision of the original feature set model is higher its Recall value is not as good as the others. There is always a trade-off between the Precision and Recall values, but it is essential to keep both of them as high as possible. We can see that the Recall values of our experiments with feature tuning, when compared with the original baseline feature set, have increased significantly more, almost 28% increase than the decrease in the Precision values of only 6.8%.

We also compared the performance of DCRFs with the baseline approach, Support Vector Machine (SVM), Bayes Net and Naïve Bayes, few of the well known classification algorithms. We used the Weka implementation of these algorithms and tested them with the same final set of features and class labels as was used for DCRFs

[25]. Table 3 shows the comparison result of our DCRFs model with the baseline approach, Bayes Net, SVM, and Naïve Bayes algorithms. The Baseline results shown in Table 3 are the same as mentioned in Figure 6 and the row representing the DCRFs result is the result of the best combination of features and class labels mentioned in Table 2 (combination of classes BME, RE and NA).

Table 3. Comparison results with the baseline approach and another machine learning approach

	Precision	**Recall**	**F1-measure**
DCRFs	0.81	0.63	0.71
SVM	0.68	0.75	0.67
Bayesian network	0.45	0.66	0.53
Naïve Bayes	0.38	0.71	0.49
Baseline	0.23	0.80	0.35

4.2 Ranking Comparison

In this section we show the comparison results of our model's ranking with those of Google's and PubMed's ranking. As mentioned in Section 3.1 of data collection, we downloaded a different set of papers from PubMed for 5 different query genes and kept the ranking given by PubMed to each of those papers. These papers acted as our repository for testing against our model as well as for indexing and ranking for Google. We gave the same set of 100 papers for each query to Google's custom search engine to index and rank. These papers were also tested against our model created in the first part of our experiment and later scored and ranked with our scoring scheme mentioned in Section 3.6. To test the ranks of all the three search results, we first needed to manually know the ground truth of these papers. Therefore, we collected the ground truth of these papers and divided them into three categories – Relevant, Not-main and Irrelevant. Table 4, shows the ground truth for all the papers for each query set, which are the different gene names. The relevant category in the table is referred to as category 1 and contains the total number of papers that have one of their main results as the coexpression of the query gene with other genes. Not-main category is category 2 and contains the total number of papers that do mention the coexpression of the query genes but not necessarily as their main result. Finally, Irrelevant category is the 3rd category with the papers that do not talk about coexpression of the query genes but may contain the coexpression word or the query gene names in them. Each column of Table 4 represents the ground truth results for each different query gene.

Once we have the ground truth, we can confidently say that in an ideal retrieval system which of the papers should come in the top retrieval result list and which ones should come at the bottom.

Table 4. Ground truth of all five query genes

		Gene names				
		Bcl2	**ErbB**	**IL**	**Myc**	**p53**
Category	**(Relevant) 1**	59	71	46	53	55
	(Not-main) 2	15	17	21	23	18
	(Irrelevant) 3	30	11	33	23	26

To compare the ranking results of our model with Google and PubMed, we used a well know evaluation metric known as Mean Average Precision (MAP) [5]. It calculates the mean of the Average Precisions (AveP) of the system on each of the query term. AveP is a measure, which computes the relevancy of the document at each retrieval step. It gives better results for a system that has relevant documents ranked at a higher rank. Equation 4 gives the formula for computing AveP for each retrieval result.

$$AveP = \frac{\sum_{r=1}^{N} \left(P(r) \times rel(r)\right)}{\#\text{ of relevant documents}}. \quad (4)$$

where r is the rank, N the number of documents retrieved, $rel()$ a binary function on the relevance of a given rank, and $P(r)$ the precision at a given cut-off rank r, given by Equation 5 below:

$$P(r) = \frac{\left|\{\text{relevant retrieved documents}\}\right|}{r}. \quad (5)$$

Finally, if we have a set of queries, the MAP is the sum of all the average precisions (*AveP*) for each query divided by the total number of queries. This is given by Equation 6 below:

$$MAP = \frac{1}{Q}\left(\sum_{q=1}^{n} AveP\right). \quad (6)$$

where Q is the set of queries from $q = 1$ to n and *AveP* is the Average Precision for each of the query term.

As we have three different categories, we want to compare the ranking results for each of the different categories individually. Hence, for each category, the other two categories were considered as totally irrelevant documents while calculating *AveP*. In this way, if a category 2 paper is wrongfully retrieved and placed in category 1, it will be treated the same as if a category 3 paper is wrongfully included in category 1. Although, this is a simple approach, it does not take into account any inter-category

relationship among categories 1, 2 and 3. It should be noted that categories 1 and 2 are much more related to the query term as they both include the papers that talk about coexpression of the query gene, even though in category 2 it is not the main subject of the paper. Category 3 is the most irrelevant category in our case because it contains the papers that either talk about coexpression of some other genes or have the query terms in them but not really the coexpression of the query gene.

Therefore, to identify with this inter-category relationship, we prepare a modified AveP that can better evaluate the quality of ranking results. This can be done by changing the value of *rel*() function from just being binary to having different weights for different category misplacement in the retrieval results. The *rel*() function in the original *AveP* equation gives 0 weight to all misplaced papers and 1 to correctly placed ones. In the Modified Average Precision (*MAveP*) the *rel*() is no longer a binary function, but has 4 different values depending upon which category a retrieved paper is placed in and which category it truly belongs to. Table 5 shows the different values we assign to the *rel*() function. The first column and the first row of the matrix are the categories of the paper returned by the ground truth and those returned by the search system and rest of values in the matrix are the values assigned to *rel()*. We can see from the table that the weight assigned to a mismatch between categories 1 and 2 is larger than that between categories 1 and 3. And the weight assigned to a mismatch between categories 2 and 3 is slightly larger than that between categories 1 and 3 but smaller than that between categories 1 and 2. If a paper is placed in its correct category, the *rel*() function will return a weight 1. The smaller the weight, the less related are two categories.

Table 5. Different values of rel() function for MAveP

		Ground truth categories		
		1	**2**	**3**
Predicted categories	**1**	1	0.75	0.25
	2	0.75	1	0.5
	3	0.25	0.5	1

We compared the ranking results in each category among our model, PubMed and Google using both AveP and MAveP. Due to space constraint we are not showing the AveP and MAveP scores for each query gene, but it was seen that for most of the

query terms and categories our model outperformed both Google and PubMed except for *IL coexpression* query term in which Google outperformed our model.

We present the MAP and Modified-MAP (Mean of MAveP) comparison results, which are the average performance of the three search systems on all the five query terms, in Tables 6 and 7. It is quite clear from the values that although Google is close to our model in categorization, still our model has a better retrieval result. This high MAP score of our model is attributed to our model's capability to better distinguish between the positive and non-positive papers. The proposed model just does not search only the paper for the occurrence of query terms, but looks for the semantic meaning of the query terms in the paper.

Our DCRFs model has been trained for tagging all the sentences that talk about coexpression of genes even though those sentences do not contain the word "coexpression" itself. And hence when a user searches for the coexpression of a particular gene with the other by providing query term as "gene name + coexpression", while other search engines try to find these two words in papers, our search engine looks for all possible ways it can be written and extracts the results that are more semantically related to the query.

Table 6. MAP comparison results for the three search systems

Category	PubMed	Google	Our Model
1	0.36	0.65	**0.7**
2	0.05	0.09	**0.14**
3	0.09	0.24	**0.31**

Table 7. Modified-MAP comparison results for the three search systems

Category	PubMed	Google	Our Model
1	0.47	0.76	**0.78**
2	0.13	0.2	**0.31**
3	0.15	0.32	**0.38**

5 Conclusion

In chapter we present a framework for extracting and ranking papers containing predicates that state coexpression relationships among genes. We trained a DCRFs model based on semantic analysis of text to classify papers that talk about gene-gene

coexpression relationships. Later, we devised a scoring scheme to score and rank papers for 5 different query terms and compared them against Google and PubMed.

There has been a lot of work done towards relationship extraction from biomedical literature, but the work presented here accomplishes more than just information extraction. In this work, we not only present a model that can extract a specialized relationship from these papers written in natural language but also a retrieval system that can be trained on this model to index and rank papers according to the importance of information present in them with regards to this specialized relationship, i.e. coexpression. Our complete framework helps in building a unified information resource that can help a researcher to get all the information needed regarding coexpression of any given gene with ease. In addition, our classification model has been trained on different ways the coexpression relationships can be expressed in literature, and so while retrieving the papers it not only retrieves the words present in the query but also all possible synonyms of those words too. This work can be extended to other domains that involve classification and ranking of unstructured data. One of the important steps in any classification algorithm is the meaningful feature extraction and accuracy of the classifiers depends upon the type of features used. Also, one commonality among all the relationship classification tasks is that we need to identify the entities, the keywords, and the positive contextual words. In this work we have emphasized the use of contextual features such as the words that occur before and after the given word and also their part of speech. This helps in giving more contextual knowledge to the model. Hence, this work can be extended and most of the features used in this work can be applied directly in any text classification task. Also, the used of Conditional Random Fields help in classification of sequential input data such as sentences and have proven to perform better than Bayes Net, SVMs and Naïve Bayes as shown in our work too.

We also use dictionary matching for the query gene to get all possible synonyms and family names of that gene. Consequently, in a way we are also performing a query expansion during the retrieval. This helps us to further present a more specialized and detailed result to the user. We can see the effect of this specialization by the improvement in the ranking result that we obtain as compared to Google and PubMed.

Some of the future ideas towards further improvement in this work can be to investigate different grammatical parsing methods, like dependency parsing and use the parse trees generated as additional features to train this model. We believe that this can improve the classification results. Another direction to explore is semi-supervised learning, as it requires less human effort of labeling the ground truth needed for training a model, and may still give sufficiently high accuracy.

This work can be considered as a step towards the semantic based information retrieval systems that can help researchers to extract accurate and relevant information from the vast array of textual information.

References

1. PubMed, http://www.ncbi.nlm.nih.gov/pubmed (1981)

2. MEDLINE Factsheet", http://www.nlm.nih.gov/pubs/factsheets/medline.html
3. Coulibaly, I. and Page, G.P.: Bioinformatic tools for inferring functional information from plant microarray data II: analysis beyond single gene. Int. J. Plant Genomics 2008 (2008)
4. Tiwari, R., Zhang, C., Solorio, T.: A Supervised Machine Learning Approach of Extracting Coexpression Relationship among Genes from Literature. In: 11th IEEE International Conference on Information Reuse and Integration, pp. 98--103 (2010)
5. Sutton, C., McCallum, A., and Rohanimanesh, K.,: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. J. Machine Learning Research 8, 693--723 (2004)
6. Voorhees, E. M.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: 21st Annual Int. ACM SIGIR, pp. 315--323 (1998)
7. Cohen, A., Hersch, W.: A survey of current work in biomedical text mining. Briefings in Bioinformatics 6, 57--71 (2005)
8. KDD 2002, The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, http://www.sigkdd.org/kdd2002 (2002)
9. Rau, L. F., Jacobs, P. S., Zernik, U.: Information extraction and text summarization using linguistic knowledge acquisition. J. Information Processing & Management **25** 4, pp. 419--428 (1989)
10. Craven, M.: Learning to extract relations from medline. In: AAAI-99 Workshop on Machine Learning for Information Extraction (1999)
11. Clark, J., Koprinska I., Poon J.: A neural network based approach to automated e-mail classification. In: IEEE/WIC International Conference on Web Intelligence, pp. 702--705 (2003)
12. Seymore K., McCallum A., Rosenfeld R.: Learning hidden markov model structure for information extraction. In: AAAI Workshop on Machine Learning for Information Extraction, pp. 37--42 (1999)
13. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. Introduction to Statistical Relational Learning, MIT Press (2006)
14. Wei, X., Croft, B., McCallum, A.: Table extraction for answer retrieval. J. Information Retrieval 9 (5), pp. 589--611 (2006)
15. McCallum A., Li W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: 7th Conference on Natural Language Learning (CoNLL), pp: 188-191 (2003)
16. Bundschus, M., Dejori, M., Stetter, M., Tresp, V., Kriegel, H. P.: Extraction of semantic biomedical relations from text using conditional random fields. J. BMC Bioinformatics 9 (207), (2008).
17. Peri, S., Navaroo, J.D., Kristiansen, T.Z., Amanchy, R., Surendranath, V., Muthusamy, B., Gandhi, T.K., Chandrika, K.N., Deshpande, N., Suresh, S., et al.: Human protein referncee database as a discovery resource for proteomics: J. Nuclein Acids Res, (2004)
18. Miyao, Y., Sagae, K., Saetre, R., Matsuzaki, T., Tsujii, J.: Evaluating contributions of natural language parsers to protein-protein interaction extraction. J. Bioinformatics 25 (3), pp. 394--400 (2009)
19. Miwa, M., Saetre, R., Miyao, Y., Tsujii, J.: A rich feature vector for protein-protein interaction extraction from multiple corpora. In: Conference on Empirical Methods in Natural Language Processing (EMNLP '09), pp. 121--130 (2009)
20. Bunescu, R., Mooney, R., Ramani, A., Marcotte, E.: Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from Medline. In: Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis (BioNLP '06), pp. 49--56 (2006)
21. Blaschke, C., Valencia, A.: The Frame-Based Module of the SUISEKI Information Extraction System. J. Intelligent Systems, 17 (2), pp.14--20 (2002)
22. Apache PDFBox – Java PDF library, http://incubator.apache.org/pdfbox/index.html

23. GENIA Tagger 3.0, www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger.
24. C. Sutton: GRMM: GRaphical Models in Mallet, http://mallet.cs.umass.edu/grmm 2006
25. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA data mining software: An update. *SIGKDD Explorations* 11(1), (2009)