A Data Mining Method to Extract and Rank Papers Describing Coexpression Predicates Semantically

Chengcui Zhang, Richa Tiwari, Wei-Bang Chen
Department of Computer and Information Sciences,
University of Alabama at Birmingham, Birmingham, Alabama, USA
{zhang, rtiwari, wbc0522}@cis.uab.edu

Abstract—**Information management and extraction in the field of biomedical research has become a requirement with the rapid increase in the amount of data being published in this area. In this paper, a graphical model, Conditional Random Fields has been used to extract a particular gene-gene relationship called "coexpression" from the existing literature. First, a Conditional Random Fields based model has been trained and tested on full-length papers downloaded from PubMed, to label the predicates that talk about coexpression of genes. Proper local and contextual text features at both word and sentence levels are proposed and extracted during the pre-processing step. The classification performance of the model trained based on the proposed features has been compared with the that of Support Vector Machines, Nearest Neighbor with generalization, and Neural Networks algorithms, and seen to outperform them all. In our second experiment, the proposed ranking scheme, which is based on classification results, is applied to the ranked lists of papers returned by PubMed and Google, respectively. The comparison of our ranked results to that of PubMed and Google demonstrates that our proposed ranking scheme performs better than both in distinguishing a positive paper from a negative paper. In conclusion, this paper describes a specialized classification and ranking framework that can retrieve papers that really talk about coexpression between and among genes based on mining of semantics and not just lexical search.**

Keywords- Conditional Random Fields; text mining; gene coexpression; relationship extraction; PubMed

I. INTRODUCTION

There is an overwhelming volume of information in the form of published papers, in the field of biomedical research, available on the Internet. This has required the researchers to develop increasingly sophisticated information management and retrieval tools. Advance techniques in the area of information retrieval help in not only retrieving the facts stated explicitly in the papers but also help in finding implicit information present in the paper. Some such information that is useful to extract not only from the experiment data but also from the published papers is gene-gene interactions, gene-disease interactions, etc. In this paper, we have focused on gene coexpression relationship extraction from the text of the literature. Two or more genes are considered to be coexpressed if they are concurrently transcribed into mRNAs, and further, translate to proteins. These genes may show functional or temporal relationships and could be useful in finding pathways and various biological processes. There are several coexpression analysis tools available for analyzing the microarray data and detecting the coexpressed genes [1]. However, there is no existing tool that can detect predicates of coexpressed genes from published articles based on text retrieval. Search engines like Google can only perform lexical searches based on the query term. However, a specialized retrieval tool which can perform semantic searches for any predicate which asserts or implies coexpression between genes is desired. In this study, we design a text mining based model for retrieving published literature with regard to gene coexpression by recognizing the relation between genes, which would greatly benefit a more comprehensive understanding of gene coexpression hypothesis.

In the past, researchers have used various techniques like Natural Language Processing, Neural Networks, etc., to analyze and form text based hypothesis [2, 3]. Recently, machine learning using graphical models such as Conditional Random Fields (CRFs), and Hidden Markov Models (HMM) have been proven successful in learning and extracting biomedical relationships [4]. CRFs are undirected graphical model developed by J. Lafferty et al. in 2001 and are preferred over HMM as they relax the input independence assumptions thus introducing rich global input features to train models [5]. CRFs have been applied to various text mining tasks, such as table extraction and biomedical named-entity extraction [6, 7]. In one of the recent works, Bundschus et al. use linear chain CRF to extract gene-disease relationship from biomedical literature and GeneRIF (Gene Reference Into Function phrases) [8].

In this work, we have used CRFs to train a model to determine the predicates that inform about the coexpression relationship between and among genes in an article. Proper local and contextual features at both word and sentence levels are proposed and extracted during the preprocessing step. We have further compared our classification results with three other data mining algorithms – Support Vector Machines (SVM), Nearest Neighbor with generalization (NNge), and Neural Networks [9, 10].

We have also searched PubMed and Google using the same query terms, applied our mining algorithm to the top

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.53

returned papers, ranked them using our proposed ranking scheme, and finally compared the rankings of Google's and PubMed's retrieval results with that of our ranking model. The comparison of our ranked results to that of PubMed and Google indicates that our proposed ranking scheme performs better than both in distinguishing a positive paper from a negative paper, with negative papers being the ones that do not talk about any particular genes being coexpressed.

In conclusion, this paper describes a specialized classification and ranking framework that can retrieve papers that really talk about coexpression between and among genes based on mining of semantics and not just lexical search.

The rest of the paper is organized as follows: Section II introduces the proposed method. Sections III and IV describe and discuss the experimental results, respectively. Section V presents the conclusions.

II. THE PROPOSED METHOD

Gene coexpression information can be useful in forming new unexpressed theories. Also, any additional information such as how, when and why these genes were coexpressed can be an added advantage. In this paper, we have described a model based on CRFs that extracts and labels the sentences containing gene coexpression predicates. CRFs are graphical models that are powerful in modeling the variables that are interdependent, for example, sequential data like text. This helps in labeling words based on not just their own features but also contextual or semantic features. An example can be a model that learns "proper nouns". The entire training and testing process can be divided into 5 steps: (1) data collection, (2) preprocessing, (3) feature extraction and class label assignment, (4) training & testing (classification), and (5) scoring and ranking.

A. Data Collection

We collected our data of positive and negative papers mostly from NCBI PubMed, one of the largest online libraries of published papers in the field of biomedical research. In total there are 510 full length papers tested against our model. These papers are manually divided into sets of positive and negative papers depending upon whether the paper conveys coexpression among genes or not. In our work, we consider the papers that talk about general coexpression, such as statistical coexpression analysis techniques, but not particular gene-gene coexpression, as negative. Fig. 1 exemplifies the distinction between positive and negative sentences. These papers are then converted into text format using Apache PDFBox Java library [11]. Once these papers were converted into text, a simple manual cleaning is performed as the PDFBox jumbles up the text in the figures or tables to the main text of the paper.

B. Text Preprocessing

The preprocessing step involves cleaning the papers with unwanted sentences that do not provide any values to the main content of the paper, such as everything under the sections 'References', 'Acknowledgements', 'Author details,' etc. We further tried to remove all the sentences that do not contain any gene names. For this purpose, we preprocess all the cleaned papers with GENIA tagger, a tool trained on Genia Corpus and PennBioIE corpus which helps in tagging the biomedical named entity [12]. Once sentences are tagged with GENIA tagger, we remove those sentences that do not contain any biomedical entities from our dataset. Therefore, each paper was represented by only those sentences that have gene names in it.

Positive Sentences

The balance between coexpressed CUC2 and MIR164A then determines the extent of serration.

Together, these data show that SphK1 is a positive regulator of MMP1 gene expression.

Thus, similar to APPswe, coexpression of X11 with APPswe Y743A retarded its maturation, prolonged its half-life, and inhibited APPs, A 40, and A 42 secretion.

Negative Sentences

The analyses show that neighboring genes are coexpressed.

Remarkably, the down-regulation in neurons expressing one functional receptor allel of both.

In this case, the coexpression relationships in the network would be robust to the choice of microarray experiments.

The authors postulate that LtpA functions as a regulator modulating the expression of genes important to B. burgdorferi's survival within its arthropod vector.

Figure 1. Examples of positive and negative sentences.

C. Feature and Class Label Assignment

This is one of the key steps in our proposed method. CRFs allow any number of input features, and the feature function can examine the entire input sequence at any point during inference. In our case, we assign local and contextual features to the words in each preprocessed and cleaned sentence as an input sequence. In particular, the training file is prepared by first parsing and tokenizing the sentences into words. Part of Speech (POS) tags represent the syntactic relationship of a word with the other words in a sentence. For example, a word can be used as a noun in a sentence and its POS tag is NN, or VB if it is a verb. Chunk tags are the POS tags for phrases, where a phrase is a group of words in the sentence that function as a single syntactic unit. For example, "a cat on the mat" is an example of a noun phrase (labeled as B-NP) and thus each word in that phrase will get a chunk tag B-NP. Table I shows some examples of the Chunk tags and POS tags for words. The root word, e.g., *coexpress* is the root word for *coexpression*, is also one of the local features. One other important local feature is the biomedical entity name tag produced by GENIA tagger. By biomedical entity name tag we refer to the entities like "gene, DNA, RNA and protein". We also assign contextual feature with a window of ±3 for each word. By a window of ±3, we refer to the 3 words and their features before and after a given word. Contextual features are very useful in machine learning for text pattern modeling and mining. In our case, contextual features can help the model in learning the relationship between the coexpression terms and the genes involved. It has been found that the term and the involved genes occur frequently within a window of ±3 and therefore we decide to use them as another feature. Consequently each word has the following features:

- Word itself
- Root word of the word
- POS tag of that word
- Chunk tag - the tag at the phrase level
- Biomedical named-entity tag
- Words and their POS and Chunk tags at position -3, -2, -1, +1, +2, +3

Once the files are prepared with tokenized and feature labeled words, we assign the class label to each token/word. For training the model, we assign '*RE*' labels to those words that represent the 'coexpression word' and '*GE*' labels to the gene names involved. The rest of the words in a sentence use their chunk tags as their class labels. We assign chunk tags but not POS tags as the default class labels for those non-RE or GE words because there are too many POS tags (on an average of 18-20 different classes), which could introduce significant noise in the feature and class label spaces. Overall we have 12 class labels from which the significant labels are the RE and GE classes.

TABLE I. AN EXMPLE OF WORDS WITH THEIR POS TAGS AND CHUNK TAGS AS THEIR LOCAL FEATURES

Word	....	POS tag	Chunk tag	...
Development		NN	B-NP	
in		IN	B-PP	
mammalian		JJ	B-NP	
cell		NNS	B-NP	
of		IN	B-PP	
a		DT	B-NP	
recombinant		JJ	B-NP	
expression		NN	B-NP	
system		NN	B-NP	

D. Training and Testing

We train our graphical CRFs model to predict class labels for words in sentences potentially expressing coexpression relationship between and among genes. In this work, we have used Mallet, a Java based McCallum et.al, implementation of CRFs [13]. A brief description of CRFs is given below.

1) Conditional Random Fields: Conditional Random Fields (CRFs) are discriminative probabilistic undirected graphical model in which each vertex is a label or output variable y that we want to infer. The edges represent the dependencies among y's, and the input variable x are the observed knowledge that can be useful in inferring the output. In CRFs, we are concerned with conditional distribution $p(y|x)$, that is, determining the conditional probability of label y given input x and some feature function, for classification. Equation (1) shows the representation of CRFs [14].

$$p(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right\} \quad (1)$$

where f_k is the transition feature function, which is equivalent to transition probabilities in HMM; λ_k is the learned parameter vector in the model; $Z(x)$ is the normalization function; K is the number of feature functions. We prepare each of our training file with correct RE and GE labels assigned to the corresponding words and train the CRFs model on 1,481 sentences, out of which 899 are positive and 582 are negative. Once the model is trained, we prepare the testing files similar to the training file but without the class labels, and test them against the trained model.

E. Scoring and Ranking

In this step we describe our proposed method to analyze and rank published literature based on the potential coexpression predicates contained in the papers. A typical query example would be – given the search terms "p53 coexpression", find all the papers that talk about p53 being coexpressed with other genes and rank them according to their relevance scores. There are three scoring techniques proposed and compared in this study as described below.

1) Scheme 1 - based on relevant sentences and their locations in the paper: In this scheme we score the paper according to the total number and location of the relevant sentences labeled by our model. We simply assume that the sentences that occur in 'abstract', 'result' and 'conclusion' weigh more than the sentences that occur in 'method' and 'materials' section. Therefore, these portions of the paper are weighed more than the others. Consequently, we calculate a weighted sum of the sentences for each paper and normalize it by the total number of pages in the paper. Equation 2 presents this scoring scheme, where $s(i)$ is the total number of relevant sentences found at location i ('abstract', 'result', 'conclusion', or 'other') and $w(i)$ is the weight for that location.

$$score = \frac{\sum s(i) \times w(i)}{t} \quad (2)$$

2) Scheme 2 - based on gene names and relevant sentences: In this scheme, we collect the number of times the query gene term occurs in the paper and also add it to the score of the previous scoring scheme. In particular, we calculate the total number of query gene term labeled by our model as GEs and divide it by the number of pages in the paper. This score will be added to the relevance scored calculated by Equation (2).

3) Scheme 3 - based on relevant genes in relevant sentences: Our third scheme is based on the number of occurrences of the query gene term in the relevant coexpression sentences (sentences with both GE and RE labels). The relevance score in this approach is calculated as the sum of score calculated by Scheme 1 and the number of relevant genes found in all the relevant sentences in that paper.

Once scores are calculated and assigned to each paper, they are ranked from high to low.

III. EXPERIMENTAL RESULTS

In the first experiment, we present the classification results of the trained CRF model on our own collection of papers sentence-wise and compare it with other data mining algorithms. In the second experiment, we apply our proposed scoring schemes on the search results (papers) returned by PubMed and Google, respectively, and compared our ranking results with that of PubMed and Google.

A. Sentence-wise Classification Result of our Proposed Model

In this section we present our labeling result and compared it with Support Vector Machine (SVM), Nearest Neighbors (NNge), and Neural networks. We started with a total of 500 papers (284 positive and 216 negative), and cleaned them by extracting the sentences with gene name to represent a paper. Our data size at the sentence level is sufficiently large (a total of approximately 9,566 sentences) for training and testing purposes, although at the paper level we did not collect too many papers because of the amount of manual work involved in preparing each paper. We trained and tested the proposed model on a data set with 3,573 positive sentences and 5,993 negative sentences, and performed 4-fold cross validation. These sentences are selected by a simple sampling technique. We divide our total set of papers into sets with equal number of papers and then randomly select equal number of papers from each set.

Once we have the sentences ready with ground truth labeled, we prepare our training files and train our model to label words expressing "coexpression" and the genes involved in the process. The ratio of positive and negative examples in each training set is approximately 1, and the ratio of training set to testing set, in each round of validation, is 2:1. To compare our results with other algorithms, we supply the same data set to the 'Weka' implementation of SVM classifier, NNge, and Neural Networks. Table II shows the F-measure value for each algorithm. It can be seen that our model outperforms all the other three classification algorithms. Our model based on CRFs performs almost 30% better than SVM in detecting coexpression predicates.

These results were quite encouraging and led us to try experimenting with the real world online search engines, and the result of our other experiments is presented in next section.

TABLE II. COMPARISON OF OUR MODEL WITH OTHER CLASSIFICATION ALGORITHMS

Classification Algorithms	Precision	Recall	F-measure
Our Model	0.667	0.490	0.565
SVM	0.534	0.317	0.398
NNge	0.381	0.379	0.380
Neural Networks	0.508	0.088	0.150

B. Paper Ranks Comparison with PubMED and Google

To compare our rankings of the papers with the state of the art, we collected NCBI and Google search results for a randomly selected search keyword in the form "gene name" + "coexpression". In this study, we only tested on two gene names, "p53" and "ErbB2". There is no particular reason these two are selected other than that we want to make sure the search engine returns enough paper (at least 50) for each term so that we can perform comparison study in a sound way.

We gathered the top 100 papers retrieved by PubMed with our keyword and manually checked for their ground truth. Due to unavailability of some of the papers, we were only able to process 82 and 85 papers for "p53 coexpression" and "ErbB2 coexpression", respectively. Also to ensure that we draw on the same set of papers to be retrieved by Google, we used Google's custom search engine to index and search papers in our dataset. Afterwards, we provide the same search term (e.g., "p53 coexpression") as we used on PubMed, to the Google custom search engine and retrieve a new ranked list of the same set of papers.

We then divide the ground truth into three groups- *Relevant*, *Not-Main*, *and Irrelevant,* and assign a numeric value to each category – 1, 2 and 3 respectively. Relevant (T) label is assigned to papers that talk about coexpression of the query gene term and that is the main result of the paper and/or is presented in the *abstract*, *introduction*, and *result* section of the paper. Not-Main (NM) is the category given to the papers that talk about the coexpression of the searched gene but not as the main result of the paper. Irrelevant (F) are the ones that are not relevant to the coexpression of the query gene. Each returned paper has gone through the classification based on our trained model, and labels for sentences are obtained this way. We then score each paper according to the three scoring schemes as mentioned in the previous section. In addition, we devise two measures, position-based and label-based measures, to evaluate and compare ranked lists.

1) Measure 1 - Position Based: Following the abovementioned labeling schemes, we are able to assign a predicted category label (1, 2, or 3) to each paper retrieved by PubMed, Google, and our schemes according to that paper's position in ranked lists. For example, if a paper was returned as the 24th paper using one of the scoring schemes for the search term 'p53 coexpression', it will be assigned a predicted category label 1 because there are 50 positive papers in the top 100. Once all the papers in a ranked list are labeled according to their positions in that ranked list, we can assign a penalty score to each by comparing the predicted label of that paper with its ground truth label. Therefore, we call this Position-based Measure.

If according to the ground truth, a paper is labeled 1 (relevant) and our ranking result gives the paper label 2 or vice-versa then a penalty of -0.25 is applied. If the ground truth is label 2 and our result gives a label 3 or vice-versa, the penalty is -0.5. The penalty is -1 for all the other cases. If both labels match, there is a reward of +1. We give higher penalty when a category 2 paper is wrongfully predicted as category 3 or vice versa, due to the fact that relevant and the not-main (1 and 2) categories are semantically closer to each other than the irrelevant to not-main category (3 and 2). Traditional measures, such as F-measure, are not suitable for this purpose since it cannot explicitly catch the relative

distance between classes. According to this measure, the higher the total score, the better the ranking result.

Table III shows the comparison results of different ranking schemes (PubMed, Google, and our proposed ranking with Schemes 1-3) using the position-based measure, given the search key term "p53 coexpression." Row 1 in the table shows the results of all the schemes in ranking relevant papers. Similarly, row 3 shows results of different schemes in ranking negative papers. Row 2 and row 4 show the results of ranking "relevant and not-main" and "not-main and irrelevant" papers, respectively. We can see that our proposed Scheme 1 significantly outperforms PubMed as well as Google in distinguishing the negative papers. Schemes 1 and 3 perform the best in distinguishing non-positive (categories 2 and 3) papers. We further compare our results with PubMed's results and Google's results in terms of the total score (the maximum score for search key term 'p53 coexpression' is 82 and for 'ErbB2 coexpression' is 85, i.e., the returned ranked list perfectly matches the ground truth ranked list in terms of their category labels) using position-based measure, as presented in Table IV. We can observe that our Scheme 3 has the highest total score of 23 for 'p53', and Scheme 2 has the highest total score of 42 for 'ErbB2'.

2) Measure 2 - Label Based: Our second measure to evaluate ranked lists of papers is referred to as label-based measure. In this design, we assume that all papers retrieved by PubMed and Google are considered to be in the relevant category and have been assigned label 1. This assumption is not unreasonable because only those papers labeled as relevant by these two search engines can be presented to the user. However, this assumption does not consider the degree of relevance of individual papers and will introduce bias in label-based measure when a dataset contain more relevant papers (according to the ground truth) than non-relevant. The labels referenced here have the same meanings as that mentioned above for the position-based measure. Once we score and label the paper, we compare our result with the labeled ground truth and assign penalties as mentioned in Section III.B.1. Table V shows the total scores of various schemes used in comparison. We can see that our Scheme 3 performs the best in ranking papers for 'p53'. For 'ErbB2', all the three proposed schemes outperform Google, and we believe that the highest score of PubMed is due to the high proportion (73%) of relevant paper in the dataset. More discussion can be found in the next section.

TABLE III. SCORE COMPARISON FOR DETERMINING THE CATEGORY OF A PAPER

Category	PubMed	Google	Scheme 1	Scheme 2	Scheme 3
1	23.5	**27.5**	23.75	21.75	24.25
1&2	22	**23.25**	20.25	20.75	22
3	-5	-2	**2.5**	-0.5	1
2&3	10.5	15	**21.75**	18.75	**21.75**

TABLE IV. POSITION BASED TOTAL SCORE COMPARISON

Term	PubMed	Google	Scheme 1	Scheme 2	Scheme 3
p53	17	21.25	22.75	20.25	**23**
ErbB2	17.5	29.5	37.5	**42**	38

TABLE V. LABEL BASED TOTAL SCORE COMPARISON

	PubMed	Google	Scheme1	Scheme 2	Scheme3
p53	30.75	30.75	27.50	27.70	**31.45**
ErbB2	**45.50**	39.45	41.00	40.50	42.00

IV. DISCUSSIONS

In this study, we are working towards building a framework that can extract papers that are specific to particular gene-gene coexpression. We predict this by considering that a sentence is positive about coexpression, if it has both gene labels (*GE*) and relation labels (*RE*). Regarding this goal, we not only have to train a model for coexpression words, but also need a mechanism to identify gene names involved in the coexpression. Training CRFs model for recognizing biological entities like gene names and protein names is a very specific task and involves its own learning together with the new set of features, like "orthographic features". Therefore, we choose to use "GENIA tagger" to tag the biomedical entities, distinguish gene names from other words, and use them as features. This does help us in determining the gene names, but we still do not have any distinct way to distinguish regular gene names from specific gene names that are being coexpressed when they are mentioned simulataneously in a paper, other than their contextual features. A 'specific coexpressed gene' referred to here is a gene that appears in the same sentence in which the coexpression term is labeled. Whereas a 'regular gene' is a gene name that occurs anywhere else in the paper.

There is a need for discriminating regular genes from coexpressed genes, based on the assumption that all the gene names that GENIA tagger identified are correct, which is not always 100% accurate. One more reason of misprediction of gene names lies in the fact that while converting from PDF to Text format, papers sometimes get garbled, and a connector such as an underscore or dot is often inserted in between words, making a word look like a gene name and not a proper English word. One example that we noticed is that GENIA tagger tagged words like "Fig. 1" as a gene name.

When we compared our results to PubMed and Google in Tables III to V, we can see that, in most cases, our proposed Scheme 3 acquires higher overall scores than both PubMed and Google using the two objective measures. This is owing to the fact that our method can distinguish between positive papers and negative papers better than those two existing search engines. We can claim this because our model is trained on finding all possible ways to express 'coexpression' and not just lexical search. In Table V, Scheme 3 yields the second highest score for ranking the 'ErbB2' papers when the label-based measure is used. PubMed achieves the highest score in this case, probably due to the bias in the label-based measure introduced by a high

proportion of relevant papers in the dataset. Further, ErbB2 has several aliases, e.g., HERB2, which can be recognized by PubMed, but not by any general search engine like Google or our current framework. We also argue that the position-based measure is more reasonable than the label-based measure because it considers the relative relevance of individual papers.

Scheme 2 scores a paper based on relevant sentences and the total gene name occurrences. This sometimes does not perform well because it counts all the GE labels tagged by our model and thus if a paper talks about the query gene anywhere other than in the relevant sentences, it will still be counted in scoring. This increases the score of the paper unnecessarily and could rank an irrelevant or not-main category paper higher. Scheme 3 is based on relevant sentences as well as the occurrences of relevant genes in relevant sentences, and it gives a better result compared with the other two schemes.

V. CONCLUSIONS

Relationship extraction from biomedical literature is of high demand and a continuously developing field. It helps in managing and easy retrieval of research outcomes on relationship study from a huge amount of published data. Coexpression relationship extraction among genes helps in identifying many theories like identifying functional properties of the genes, analyzing pathways, etc. In this paper, we propose and demonstrate a framework to identifying and ranking papers that talk about coexpression from the search results returned by PubMed and Google, by using a graphical model (CRFs) based data mining method to label predicates describing coexpression. We compared our ranking results with that of Google and PubMed on two search terms and found that our model outperforms both of them in distinguishing between negative and positive papers. Finally, we conclude that our model is a specialized search framework that can retrieve and rank relevant papers from biomedical literature for extraction of coexpression of genes based on semantic features and not just by lexical search.

ACKNOWLEDGMENT

The research of Chengcui Zhang is supported in part by NSF DBI-0649894 and UAB ADVANCE program.

REFERENCES

[1] I. Coulibaly and G. P. Page, "Bioinformatic tools for inferring functional information from plant microarray data II: analysis beyond single gene," International Journal of Plant Genomics, vol. 2008, Jul. 2008, pp. 1-13, doi: 10.1155/2008/893941.

[2] C. B. Giles and J. D. Wren, "Large-scale directional relationship extraction and resolution," BMC Bioinformatics, vol. 9 (Suppl 9), Aug. 2008, pp. S11, doi: 10.1186/1471-2105-9-S9-S11.

[3] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in Proceedings of IEEE/WIC International Conference on Web Intelligence, Halifax, Canada, Oct. 2003, pp. 702-705.

[4] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in Proceedings of AAAI Workshop on Machine Learning for Information Extraction, 1999, pp. 37-42.

[5] C. Sutton, and A. McCallum, "An introduction to conditional random fields for relational learning," Introduction to Statistical Relational Learning. MIT Press, 2006.

[6] X. Wei, B. Croft, and A. McCallum, "Table extraction for answer retrieval," Information Retrieval Journal, vol. 9 (5), Nov. 2006, pp. 589-611, doi: 10.1007/s10791-006-9005-5.

[7] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in Proceedings of the Seventh Conference on Natural Language Learning (CoNLL), Edmonton, Canada, 2003, pp. 188-191, doi: 10.3115/1119176.1119206.

[8] M. Bundschus, M. Dejori, M. Stetter, V. Tresp, and H. P. Kriegel, "Extraction of semantic biomedical relations from text using conditional random fields," BMC Bioinformatics, vol. 9 (207), Apr. 2008, doi: 10.1186/1471-2105-9-207.

[9] B. Martin, Instance-Based learning: Nearest Neighbor with Generalization, Hamilton, New Zealand, 1995.

[10] Y. EL-Manzalawy. "WLSVM", Available at http://www.cs.iastate.edu/~yasser/wlsvm/, 2005.

[11] Apache PDFBox – Java PDF library, Available at http://incubator.apache.org/pdfbox/index.html

[12] GENIA Tagger 3.0, Available at www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger.

[13] C. Sutton, "GRMM: GRaphical Models in Mallet," Available at http://mallet.cs.umass.edu/grmm, 2006.

[14] H. M. Wallach, "Conditional Random Fields: An Introduction," Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.