CTCPI – Convolution Tree Kernel-based Chemical-Protein Interaction Detection

Neha Warikoo$^{1,2,5}$, Yung-Chun Chang$^{3}$, Po-Ting Lai$^{4,5}$ and Wen-Lian Hsu$^{5}$

$^{1}$Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei, Taiwan,
$^{2}$Institute of Biomedical Informatics, National Yang-Ming University, Taipei, Taiwan,
$^{3}$Graduate Institute of Data Science, Taipei Medical University, Taipei, Taiwan,
$^{4}$Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan,
$^{5}$Institute of Information Science, Academia Sinica, Taipei, Taiwan

Abstract: In this paper we introduce a chemical-protein interaction detection system called CTCPI, which uses Convolution Tree Kernel to separate various levels of interaction mechanisms exhibited by chemicals while interacting with proteins/genes. Our system enlists a novel feature engineering method based on Algebraic Invariance to identify and consolidate distinct linguistic features for each class from the candidate sets and use these feature patterns as a screening function for generating the feature tree for SVM-CTK classifier. Our system achieved about 30.92% performance for chemical protein interaction class identification task.

Keywords: Named Entity Recognition (NER), Algebraic Invariance, Convolution Tree Kernel (CTK)

I. INTRODUCTION

Biological interactions are vital in understanding the mechanism of action involved behind the physiological response to any drug. The nature of interaction can help us ascertain what category a drug belongs to, whether it inhibits an unwarranted gene regulation or works as a conjugate in some other reaction etc. There is plethora of published material with references to drug/chemical based interactions with various biological entities which can be used in subsequent researches to develop more effective drugs or be used in clinical practices for bettering patient care (7). BioCreative VI – ChemProt task is aimed at identifying relation types targeting “chemical compound-based interaction with proteins/genes” (CPI). The task is focused on identifying true interactions in proteins/genes stimulated or caused directly by chemical compounds/drugs and annotating the respective nature of interaction between the corresponding entities (4). As per the task we are to categorize these interactions into given 5 classes viz, Class 3 – Up Regulators, Class 4 – Down Regulators, Class 5 – Agonist, Class 6 – Antagonist and Class 9 – Substrate or Product Of.

II. METHOD

The Chemical Protein Interaction task is a multi-label classification task where named entity types for each abstract in the corpus data were provided by the task organizers. To deliver the task, we developed a 4-tier module system as shown in Figure 1. Tier1 is the Candidate Instance Generation System, which is responsible for pre-processing of input abstracts. This module entails screening for relation-oriented sentences, which are referred to as “Instances” in our system. This module is preceded by generic tasks of natural text preprocessing viz. Sentence Detection (Apache Open NLP) $^1$, Entity Class Labeling (In-built Module), and part-of-speech (POS) tagging (Genia Tagger) $^2$ (8-9). Hence generated POS-tagged sentences were drafted into candidate instances based on entity pairs (one chemical and one protein/gene pair mention per sentence per pairwise iteration) relabeling to indicate the primary Chemical and Protein pair. The verb implying the relation (proximal verb) was also assigned a prominent identifier as “Relation”. It entails a non-basal form verb nearest to the current entity pair set. The use of this form can be explained by linguistic predominance in describing causal relations using a non-basal form verb in a smaller frame between the respective subject-object within the sentence.

The Tier2 of our system is based on a feature engineering method novel to the natural text domain. This module is dubbed as Algebraic Invariance-based Feature Engineering set. The module is responsible for generating a tree-based feature file where Invariance method is used for feature enrichment. In contrast to using the whole parse tree directly generated by Stanford parser (1), a set of stringent context feature patterns based on Invariance functional were used to set the upper and lower bounds for candidate instances accepted for tree generation. In addition, context patterns per instance were used for pruning and decorating the respective instance tree, thereby enhancing the feature context and bringing brevity to the features incorporated.

Our feature engineering approach is based on conjecture that different candidate instances show similarity in inference even if they are structurally diverse when relevant contexts are used as reference points. The objective is to demonstrate the invariance or lack of change in the nature of such descriptive sections from the text, and exploit this characteristic in generating more robust features while limiting the degree of

2. http://www.nactem.ac.uk/tsujii/GENIA/tagger/geniatagger-3.0.2.tar.gz
evaluation function. The idea is heavily drawn on Algebraic Invariance to show that two separate sentences are similar in their inferential meaning if their invariant function does not vary (2). Such a function can be represented as follows:

\[
I(q_{a0}...q_{0n}) = \Delta^n * I(p_{a0}...p_{0n})
\]

where \( I(q) \) and \( I(p) \) indicate the invariant function post and prior to transformation (T), \( \Delta \) is the determinant of the representation polynomial undergone transformation (T), and \( W \) is the invariant weight.

In order to restructure the invariance concept in a natural text paradigm, we assumed a homogenous polynomial function based on our prior selected three key referential labels viz. Entity1 (Chemical), Relation (Proximal Verb), and Entity2 (Protein/Gene) to project every instance in the Euclidian space. We limited our function to a second order polynomial based on each variable set as given below:

\[
P(x,y) = p_{20}x + p_{11}y + p_{02}y^2
\]

where \( x \) and \( y \) are binary association variables representing “Entity1~Relation” and “Entity2~Relation” respectively. \( p_{20}, p_{11}, \) and \( p_{02} \) are coefficients of the polynomial evaluated by the maximum value from a seven-frame adjacency matrix vector for each of the corresponding variable pairs. This seven-frame adjacency matrix is calculated for each variable in the combination set and is based on an n-gram probabilistic model by shifting the window frame iteratively over the instance, moving reference label index from 1 through 7. We settled on 7-gram model as it tested positively with the development data. Per instance the cumulative n-gram score from union of similarity instances per variable over all candidate instances is taken as statistical significance score for the context pair set and is labeled as coefficient of the variable pair.

Our algorithm treats each candidate instance polynomial as a transformed version of all other instance polynomials. Using (1), if the invariant functional of the current candidate polynomial is equal to the invariant functional of other instance polynomials, then the current instance is considered similar to each of those instances, thereby reducing the dimensionality of screening space for pattern generation and keeping context-specific similarities.

Keren, D (4) developed a method for calculating invariance functional for homogenous polynomials. Using that methodology and based on (1) and (2), we determine our invariance functional as given below:

\[
I(q_{a0}...q_{0n}) = I(p_{a0}...p_{0n}) = \left( p_{20} + \frac{p_{11}}{2} + p_{02} \right)
\]

where \( I(q) \) and \( I(p) \) are the invariant functions for the transformed instance polynomial \( Q(u,v) \) and original instance polynomial \( P(x,y) \), respectively. \( p_{20}, p_{11}, \) and \( p_{02} \) are the coefficients of the original polynomial function \( P(x,y) \). Every candidate instance is screened for the 3 key referential labels, and then corresponding coefficient values from the homogenous representation equation (2) are substituted in equation (3) to obtain the invariant function score \( I(p)_k \) in
which \( k \) is the current candidate instance ID. The instances are sorted in descending order of their respective scores. If the invariant scores of two consecutive scores are approximately similar (\( \Delta = 1.00 \) and \( W \sim 1 \), using (1)), then the instances are deemed as inferentially similar and thereby clustered together. Otherwise, the instances are diversified into separate groups. Instances within a group are aligned together to generate 7-frame size context pattern per key referential label. Consequently, we obtain per category specific screening patterns, which are used for feature enrichment in tree building. The context patterns generated are compared against the referential label-based frames from the candidate instances to screen the information content and relevance of the respective instances to be used for the classifier.

The hence generated pruned and decorated tree is subjected to Tier3 of our system module where further removal of noise from the candidate tree instances is attempted by using trigger words specific to each category. These trigger terms were identified based on statistically significant biological relational keywords obtained from the training corpus. The trigger terms are separated class wise and used to eliminate cross-referenced instances. Only filtered tree instances are allowed to be in the final feature file. The last segment i.e Tier4 of the tool is SVM classifier based on convolution tree kernel which is run on One vs One approach to segregate multiple interaction classes from each other. SVM-Light-TK-1.5\(^3\) toolkit was used in both the learning and classification modules (5-6).

### III. Experiments

**Experimental SetUp**

The dataset provided by the task organizers for separate phases varied in the abstract size. The training data extended to 1020 abstracts and 4157 interaction relations. The development data spanned about 612 abstracts with 2416 relations. Three out of five task submissions used a balanced cubed model of training and development data while for the remaining two we employed a balanced training set instances for the model learning. We trained all the models on SVM-CTK. Other than using varied sizes of training data, we used variants of tree and vector features; by employing a combination model (tree + vector, \#2), only tree-based model (\#1,3,5) and only vector-based model (\#4) for separate runs. The entity recognition was already provided with the respective corpuses, therefore our system only focused on chemical-based interaction with proteins/genes. The standard evaluation metrics of precision, recall and F1-score were used for calibrating the method efficiency.

**Results and Discussion**

Table 1 briefs the performance of our system for each of the runs submitted. A total of 5 runs were obtained for prediction on the test set. The system registered F1-Score of about 30.92% on the best run, with recall and precision measuring up to 32.71% and 29.32% respectively. The model with best run is based on a consolidated corpus using both training and development data, weighing heavily on tree-based features for classification in contrast vector features. Effectiveness of tree kernel in classification indicates that the class specific features introduced by our feature engineered tree pruning apparatus are relatively effective in enriching qualitative contribution of parse trees as opposed to using only vector features.

<table>
<thead>
<tr>
<th>Run</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.2446</td>
<td>0.3407</td>
<td>0.2847</td>
</tr>
<tr>
<td>#2</td>
<td>0.2563</td>
<td>0.3456</td>
<td>0.2943</td>
</tr>
<tr>
<td>#3</td>
<td><strong>0.2932</strong></td>
<td><strong>0.3271</strong></td>
<td><strong>0.3092</strong></td>
</tr>
<tr>
<td>#4</td>
<td>0.0729</td>
<td>0.0150</td>
<td>0.0249</td>
</tr>
<tr>
<td>#5</td>
<td>0.2587</td>
<td>0.3456</td>
<td>0.2959</td>
</tr>
</tbody>
</table>

### IV. Conclusion

This paper introduces a system dedicated to identifying multi class chemical-protein interactions from pub-med abstracts. Multi-class classification makes this task challenging than usual binary classification tasks. In addition, there is also overlap in terms of interaction expression or key term references when it comes some interaction types, which brings in further ambiguity for classification. Our system was impactful to certain extent in segregating various interaction types, however an additional resource of trigger word features from curated literature database would be crucial in noise removal and better screening for interaction types.

### REFERENCES


\(^3\) [http://disi.unin.it/moschitti/TK1.5-software/download.html](http://disi.unin.it/moschitti/TK1.5-software/download.html)
