Mining protein interactions affected by mutations using a NLP based machine learning approach

Albert Steppi[1], Jinchan Qu[1], Jie Hao[1], Jian Wang[1], Pei-Yau Lung[1], Tingting Zhao[2], Zhe He[3], Jinfeng Zhang[1,*]
[1]Department of Statistics, Florida State University, Tallahassee, FL 32306, USA
[2]Department of Geography, Florida State University, Tallahassee, FL 32306, USA
[3]College of Communication and Information, Florida State University, Tallahassee, FL 32306, USA

***Abstract*— The knowledge on the protein/gene interactions that are affected by mutations helps understand phenotype-genotype associations and predict disease prognosis and responses to treatments. Such information is scattered around in scientific literature and its manual curation is very time and resource consuming. Although much research has been done in the past to extract protein-protein interaction (PPI) information automatically from literature, much less has been done on extracting PPIs affected by mutations. An important step towards extracting such information is automatically retrieving the relevant articles. In this study, we classify abstracts as relevant or irrelevant, where being relevant means the article contains information that at least one PPI is affected by a mutation. We started out with a bag-of-words and bigrams model, and then added some frequency statistics. Interactions are modeled by triplets, where a triplet is defined as two protein names and an interaction word in a sentence. The Stanford Dependency Parser, which helps find the dependency graph for each sentence, was used to further calculate the shortest path distances for each of the three pairs of words in a triplet. We also used a model to estimate the probabilities that a triplet forms a true interaction. The shortest paths between protein name and mutation word pairs in the sentences were computed as well. Features based on all these above were used in a Gradient Boosting Trees model. Our model achieved satisfactory performance in the training data of the Biocreative VI challenge.**

Keywords*— *Protein-protein interactions; mutations; text mining; biomedical literature retrieval; protein interactions affected by mutations

I. INTRODUCTION

The publications in biomedical literature have been increasing at an accelerated speed. Reading all the literature in a particular domain of interest has become less and less feasible in recent years. And the gap between the number of papers a researcher can read and the number of all the papers he/she needs to read will only become wider over time. Searching literature using keywords at literature databases such as PubMed has been a very popular approach for finding relevant scientific articles. However, such searches can often return hundreds or even thousands of papers, many of which are not highly relevant to the topics a user aims to find. Biomedical literature triage (or retrieval) has been a popular topic in recent years [1-12]. Most of the methods use machine learning methods to retrieve relevant articles related to a particular topic. Literature triage tasks vary widely depending on the subdomain of the literature and also on the particular information one aims to retrieve.

In this study, we address the problem raised in the Biocreative challenge VI track 4, mining protein interactions and mutations for precision medicine. This challenge consists of two subtasks: (1) Document triage: identify relevant PubMed citations describing genetic mutations affecting protein-protein interactions; and (2) Relation extraction: extract experimentally verified protein-protein interactions (PPI) affected by the presence of a genetic mutation. Much research has been done in the past to extract protein-protein interaction (PPI) information automatically from literature [13-24]. Extracting PPIs affected by mutations is very challenging in that not only one needs to deal with information on PPIs, but also on whether they are affected by mutations. The information on whether a PPI is affected by mutations often occurs in sentences different from the one where PPIs are mentioned, which makes the tasks more challenging.

For document triage task, the training dataset consists of a set of ~4K PubMed articles. These articles are manually labelled as relevant/not relevant by BioGRID database [25] curators. The goal is to build automatic methods capable of receiving a list of PMIDs and return a relevance-ranked judgement of the test set for triage purposes.

For relation extraction task, a subset of the relevant articles in document triage task was manually annotated with relevant interacting protein pairs. Each PubMed article in this set has at least one interacting pair which is listed with the Gene Entrez ID of the two interactors. These protein-protein interactions have been experimentally verified and the analysis of natural occurring or synthetic mutations has identified protein residues crucial for the interaction. The goal is to build automated methods that are capable of receiving a set of PMID documents and return the set of interacting protein pairs (and their corresponding Gene Entrez IDs) mentioned in the text that are affected by a genetic mutation.

The validity of the text mining methods is evaluated using standard metrics such as average precision, f-measure, etc. In addition, we also used Area Under the Curve (AUC) to evaluate different models.

In the rest of the paper, we will first describe the method we developed for both tasks. We will then present our result on mainly the first task. We end the paper with conclusion and discussion.

This study was partially supported by a grant from NIGMS of National Institute of Health under award number R01GM126558.

II. Methods

A. Data

The data used in this work are from PubMed articles. The dataset consists of titles and abstracts from 4082 papers, each manually labeled by BioGRID database curators as relevant or not relevant, for the training data, and 1500 papers for the test data.

B. Manual curation of mutation related words

Mutation related words were first manually curated from a list of words with a high TF-IDF between the true and false labels. When using cross validation to tune the model, mutation words not included in the training fold were excluded to avoid a potential information leak.

C. Feature extraction and model building

1) Triplet based modeling for protein-protein interactions

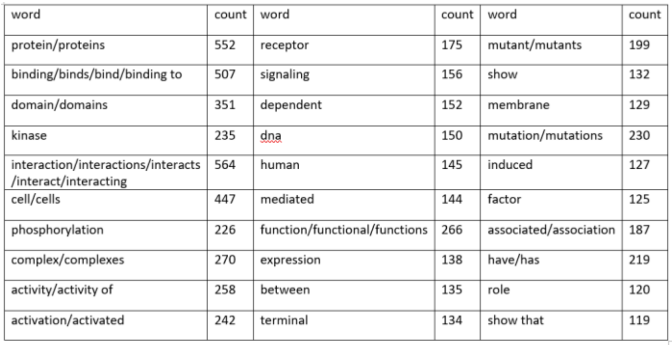
Protein-protein interactions (PPI) were modeled based on a triplet concept we developed in a previous study [26], where a triplet is defined as two protein names and an interaction word describing their interaction or relationship in the same sentence. The dictionary of interaction words were developed by a combination of manual and computational approach [26]. The method was later successfully used in knowledge discovery and other applications [27-30].

2) Bag-of-words and bigrams model

We started out with a Bag-of-words and bigrams model, and then added some frequency statistics from the title and body parts of the abstract, given that the title and the body have different information for triage. Treating the whole document as a bag, a word's potential to become a feature for training classifiers will depend on its term frequency in each abstract compared with its occurrence through the whole document. This way the words that appear often in the whole bag will be weighted less when deciding features. Besides the features from Bag-of-words, some frequency statistics were proved to be contributive in our initial Support Vector Machine model. These include: the percentage counts of protein/gene names, mutation words and interaction words, the percentages of sentences in an abstract that have 1, 2 or 3 and above pairs of protein name and mutation word within the sentence, the percentage count of within-sentence protein name and mutation word pairs in an abstract, and an indicator of whether a protein name and mutation word both appear in the title. Table 1 shows some of the most frequent words identified and kept in the model at this step.

We took this linear support vector machine (SVM) as our baseline model. It achieved an ROC-AUC score of 0.724 (std 0.0812) an F1 score of 0.5999 (std 0.05073). This linear SVM with bag-of-words features gives a baseline for our models performance.

TABLE I. MOST FREQUENT VOCABULARY AFTER BAG-OF-WORDS



word	count	word	count	word	count
protein/proteins	552	receptor	175	mutant/mutants	199
binding/binds/bind/binding to	507	signaling	156	show	132
domain/domains	351	dependent	152	membrane	129
kinase	235	dna	150	mutation/mutations	230
interaction/interactions/interacts /interact/interacting	564	human	145	induced	127
cell/cells	447	mediated	144	factor	125
phosphorylation	226	function/functional/functions	266	associated/association	187
complex/complexes	270	expression	138	have/has	219
activity/activity of	258	between	135	role	120
activation/activated	242	terminal	134	show that	119

3) Stanford Dependency Parser

The Stanford Dependency Parser was used to find the dependency graph for each sentence. For each triplet consisting of two protein names and an interaction word in a sentence, we computed the shortest path in the dependency graph between every pair of words in the triplet. Imagine each word is a node, each typed dependency is an edge that connects two nodes, then the parser can give the network graph of a sentence. Given two nodes, for example, Protein 1 and Protein 2, or Protein 1 and its interaction word with Protein 2, we need to find which path that connects these two nodes are the shortest. We can then obtained the Abstract-level features based on this information. For instance, the percentage counts of the triplets in an abstract that have the shortest path distances equaling 1, 2, 3, falling in the range 4 to 6, falling in the range 7 to 11, or equaling 12 and above, the percentage of sentences that have shortest path information in an abstract, etc.

We also used a model developed by another member of our group to extract true PPIs from the abstracts [26, 28, 31]. This model also uses information of the dependency graph generated by the Stanford Dependency Parser. Intuitively the shortest path features give us information to distinguish true PPIs among all the triplets. Triplets consisting of two proteins and an interaction word in the same sentence were then extracted from the abstracts and the model was used to estimate the probabilities that they form a true interaction. Therefore, by estimating the probabilities for triplets in an abstract to be true (PPIs), we added more features such as the percentage counts of the triplets with certain probability ranges. (We took the three equally sized bins [0, 1/3) (1/3, 2/3], (2/3,1].) We would then count the frequency of triplets with predicted probability falling into each bin.

The previous features give an idea of how many protein, interaction, and mutation words are contained in an abstract and an estimate for how many true PPIs are contained, but we have not yet done anything that connects the mutations to PPIs.

We computed the shortest paths in the dependency graphs between protein names and mutation words and added additional features based on these. For each protein word mutation word pair, we computed the shortest path between them. For each such pair, we computed the maximum probability given by our PPI extraction method over all triples containing the given protein. The probabilities were again put into bins [0 <= p < 1/3], [1/3 <= p < 2/3], [2/3 <= p <=1]. Shortest path lengths were again placed into the bins 1, 2, 3, 4-6, 7-11, 12 and above as before, then binned again according to max PPI probability. For example, suppose there is a path connecting the protein tubulin with the mutation words SNP of length 8. This path will fit into the bin [7 <= sp <=11]. If our PPI extraction model gives a maximum probability of 0.56 that a triplet containing tubulin is a true PPI then this path will go into a bin with [7 <= sp <= 11] and [1/3 < p <= 2/3] since there are three probability bins and 6 shortest path length bins, there are a total of 18 bins a path length can go into. For each abstract, we take as a feature, the frequency of shortest paths lengths in each bin. For convenience we will call the probability bins: [0 <= p < 1/3] as bin 0, [1/3 < p < 2/3] as bin 1, [2/3 <= p <= 1] as bin 2.

We also employ a bag of words and bigrams approach to the words along each shortest path and to the dependencies along each path. If a path has a max PPI probability falling into bin 2, we append 2 to the end of each word in the path as follows: Tyorosines___2 recognized___2 trapped___2.

And similarly for dependencies. This shortest path had a max PPI probability falling into bin 1, so we have appended a 1 to each value. This is done so we can distinguish between paths connecting mutation words to proteins that have interactions from paths connecting to proteins that don't have any listed interactions: Xcomp___1 dep___1 xcomp___1 acl:relcl___1 root___1 dep___1 advmod___1 acl___1.

Frequencies of protein words, interaction words, and mutation words were also computed along the shortest paths.

4) *Gradient boosting trees model*

We tested three classifier types, a Linear SVM, a Random Forest, and a Gradient Boosted Trees model. The first two implemented in Python's scikit-learn. We used the popular XGBoost library[32] for gradient boosted trees. The number of words included in the bag of words and bigrams features was limited in the XGBoost model to avoid overfitting. The models were tuned using ROC-AUC scoring. The probability scores for the best model was then calibrated using Platt scaling[33]. Then cross validation was used to find a cutoff threshold to maximize the F1 score. A cutoff of p = 0.35 was chosen to predict abstracts as relevant. The model's performance on the training data was reasonable, with a cross-validated ROC-AUC score of 0.793 (std 0.018) and F1 score of 0.709 (std 0.0090).

III. Results

The following results were computed using ten-fold cross validation. Our baseline Linear SVM model using bag of words features and some count statistics received an ROC_AUC score of 0.74276. It was found that the SVM's Performance was sensitive to the choice of transformer weights in the scikit-learn FeatureUnion transformer. These weights were tuned with cross validation. We then enriched the feature set with information from the Stanford Dependency Parser and a PPI extraction model developed in-house based on some previous studies of ours [26, 28, 31]. We tried both a linear SVM and a Random Forest to the data with these additional features. With the additional features the ROC_AUC score of the linear SVM changed very slightly to 0.7428. A random forest with 1000 estimators, min samples leaf set at 2 and a Gini splitting criterion (these parameter values were found through cross-validation) achieved an ROC-AUC score of 0.78825 with these features, giving significantly better results than the SVM. Applying a tuned XGBoost to this same set of features yielded an ROC_AUC score of 0.78686. With all features included, the best scoring model was a tuned XGBoost with an ROC_AUC score of 0.79304. In this case, the performance of the Random Forest barely improved with the addition of the new features.

IV. Conclusions and Discussions

In this study, we tackled the problem of retrieving abstracts which contains at least one protein-protein interactions (PPI) affected by a mutation. We developed natural language processing (NLP) based methods, extracted a set of diverse features, and experimented with several popular machine learning methods. The performance on the training data of Biocreative VI was quite satisfactory.

There is still much room for further improvement of our models, especially in terms of optimizing the set of features we will use for the machine learning models. These will be the subject of future studies.

Acknowledgment *(Heading 5)*

We thank the help from Yuhang Liu, Zihan Cui, and Linlin Sha for reading some of the text and interpretation of the prediction results.

References

1. Wang, J.Z., et al., *G-Bean: an ontology-graph based web tool for biomedical literature retrieval.* BMC Bioinformatics, 2014. **15**.
2. Vishnyakova, D., et al., *Utilization of Ontology Look-Up Services in Information Retrieval for Biomedical Literature*, in *Data and Knowledge for Medical Decision Support*, B. Blobel, A. Hasman, and J. Zvarova, Editors. 2013. p. 155-159.
3. Cherdioui, S. and F. Boubekeur, *Information Retrieval Techniques for Knowledge Discovery in Biomedical Literature*. 2013 11th International Symposium on Programming and Systems, ed. A. Guessoum, Y. Djouadi, and A. Djouama. 2013. 137-142.
4. Dogan, R.I. and L. Yeganova, *Topics in machine learning for biomedical literature analysis and text retrieval.* BMC Bioinformatics, 2011. **12**.
5. Zhou, X.H., et al., *Relation-based document retrieval for biomedical literature databases*, in *Database Systems for Advanced Applications, Proceedings*, M.L. Lee, K.L. Tan, and V. Wuwongse, Editors. 2006. p. 689-701.
6. Singhal, A., M. Simmons, and Z. Lu, *Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine.* PLoS Comput Biol, 2016. **12**(11): p. e1005017.

7. Kim, S., et al., *BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID.* Database (Oxford), 2016. **2016**.

8. Wei, C.H., et al., *tmVar: a text mining approach for extracting sequence variants in biomedical literature.* Bioinformatics, 2013. **29**(11): p. 1433-9.

9. Huang, C.C. and Z. Lu, *Community challenges in biomedical text mining over 10 years: success, failure and the future.* Brief Bioinform, 2016. **17**(1): p. 132-44.

10. Comeau, D.C., et al., *BioC: a minimalist approach to interoperability for biomedical text processing.* Database (Oxford), 2013. **2013**: p. bat064.

11. Wei, C.H., H.Y. Kao, and Z. Lu, *PubTator: a web-based text mining tool for assisting biocuration.* Nucleic Acids Res, 2013. **41**(Web Server issue): p. W518-22.

12. Wei, C.H., R. Leaman, and Z. Lu, *Beyond accuracy: creating interoperable and scalable text-mining web services.* Bioinformatics, 2016. **32**(12): p. 1907-10.

13. Bui, Q.-C., S. Katrenko, and P.M.A. Sloot, *A hybrid approach to extract protein-protein interactions.* Bioinformatics (Oxford, England), 2011. **27**: p. 259-265.

14. Bui, Q.-C., et al., *Extracting causal relations on HIV drug resistance from literature.* BMC bioinformatics, 2010. **11**: p. 101.

15. Ceol, A., et al., *Linking entries in protein interaction database to structured text: the FEBS Letters experiment.* FEBS letters, 2008. **582**: p. 1171-1177.

16. Chowdhary, R., J. Zhang, and J.S. Liu, *Bayesian inference of protein-protein interactions from biological literature.* Bioinformatics (Oxford, England), 2009. **25**: p. 1536-1542.

17. Giles, C.B. and J.D. Wren, *Large-scale directional relationship extraction and resolution.* BMC bioinformatics, 2008. **9 Suppl 9**: p. S11.

18. Hu, X. and D.D. Wu, *Data mining and predictive modeling of biomolecular network from biomedical literature databases.* IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, 2007. **4**: p. 251-263.

19. Huang, M., et al., *Mining physical protein-protein interactions from the literature.* Genome Biol, 2008. **9 Suppl 2**: p. S12.

20. Krallinger, M., et al., *Overview of the protein-protein interaction annotation extraction task of BioCreative II.* Genome biology, 2008. **9 Suppl 2**: p. S4.

21. Krallinger, M., F. Leitner, and A. Valencia. *Assessment of the {S}econd {B}io{C}reative {PPI} Task: {A}utomatic Extraction of Protein-Protein Interactions*. in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*. 2007.

22. Pyysalo, S., et al., *Comparative analysis of five protein-protein interaction corpora.* BMC bioinformatics, 2008. **9 Suppl 3**: p. S6.

23. Tikk, D., et al., *A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature.* PLoS Computational Biology, 2010. **6**(7): p. e1000837.

24. Krallinger, M., et al., *The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text.* BMC Bioinformatics, 2011. **12 Suppl 8**: p. S3.

25. Chatr-Aryamontri, A., et al., *The BioGRID interaction database: 2017 update.* Nucleic Acids Res, 2017. **45**(D1): p. D369-D379.

26. Chowdhary, R., J. Zhang, and J.S. Liu, *Bayesian inference of protein-protein interactions from biological literature.* Bioinformatics, 2009. **25**(12): p. 1536-42.

27. Lindsey Bell, et al., *Integrated bio-entity network: a system for biological knowledge discovery.* PLoS One, 2011. **6**(6): p. e21474.

28. Bell, L., J. Zhang, and X. Niu, *Mixture of logistic models and an ensemble approach for extracting protein-protein interactions.* ACM-BCB, 2011: p. 371-375.

29. Chowdhary, R., et al., *Context-specific protein network miner--an online system for exploring context-specific protein interaction networks from the literature.* PLoS One, 2012. **7**(4): p. e34480.

30. Balaji, S., et al., *IMID: integrated molecular interaction database.* Bioinformatics, 2012. **28**(5): p. 747-9.

31. Zhang, J., *Automatic extraction of bio-entity relationships from literature. USPTO No. 8,886,522* 2014, Florida State University: USA.

32. Chen, T. and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, ACM: San Francisco, California, USA. p. 785-794.

33. Platt, J.C., *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*, in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. 1999, MIT Press. p. 61-74.