Mining Heterogenous Relationships from Pubmed Abstracts Using Weak Supervision

This manuscript (permalink) was automatically generated from greenelab/text_mined_hetnet_manuscript@ec2bd21 on July 23, 2019.

Authors

- **David N. Nicholson**
  0000-0003-0002-5761 · danich1
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania · Funded by GBMF4552

- **Daniel S. Himmelstein**
  0000-0002-3012-7446 · dhimmel · dhimmel
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania · Funded by GBMF4552

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · GreeneScientist
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania · Funded by GBMF4552 and R01 HG010067

Abstract

This is a **rough draft** of a manscript on label function reuse for text mining heterogenous relationship from Pubmed Abstracts.

#Introduction Set introduction for paper here Talk about problem, goal, and significance of paper

Recent Work

Talk about what has been done in the field in regards to text mining and knowledge base integration

Materials and Methods

Hetionet

Hetionet [1] is a large heterogenous network that contains pharmacological and biological information. This network depicts information in the form of nodes and edges of different types: nodes that represent biological and pharmacological entities and edges which represent relationships between entities. Hetionet v1.0 contains 47,031 nodes with 11 different data types and 2,250,197 edges that represent 24 different relationship types (Figure 1). Edges in Hetionet were obtained from open databases, such as the GWAS Catalog [2] and DrugBank [2]. For this project, we analyzed performance over a subset of the Hetionet relationship types: disease associates with a gene (DaG), compound binds to a gene (CbG), gene interacts with gene (GiG) and compound treating a disease (CtD).

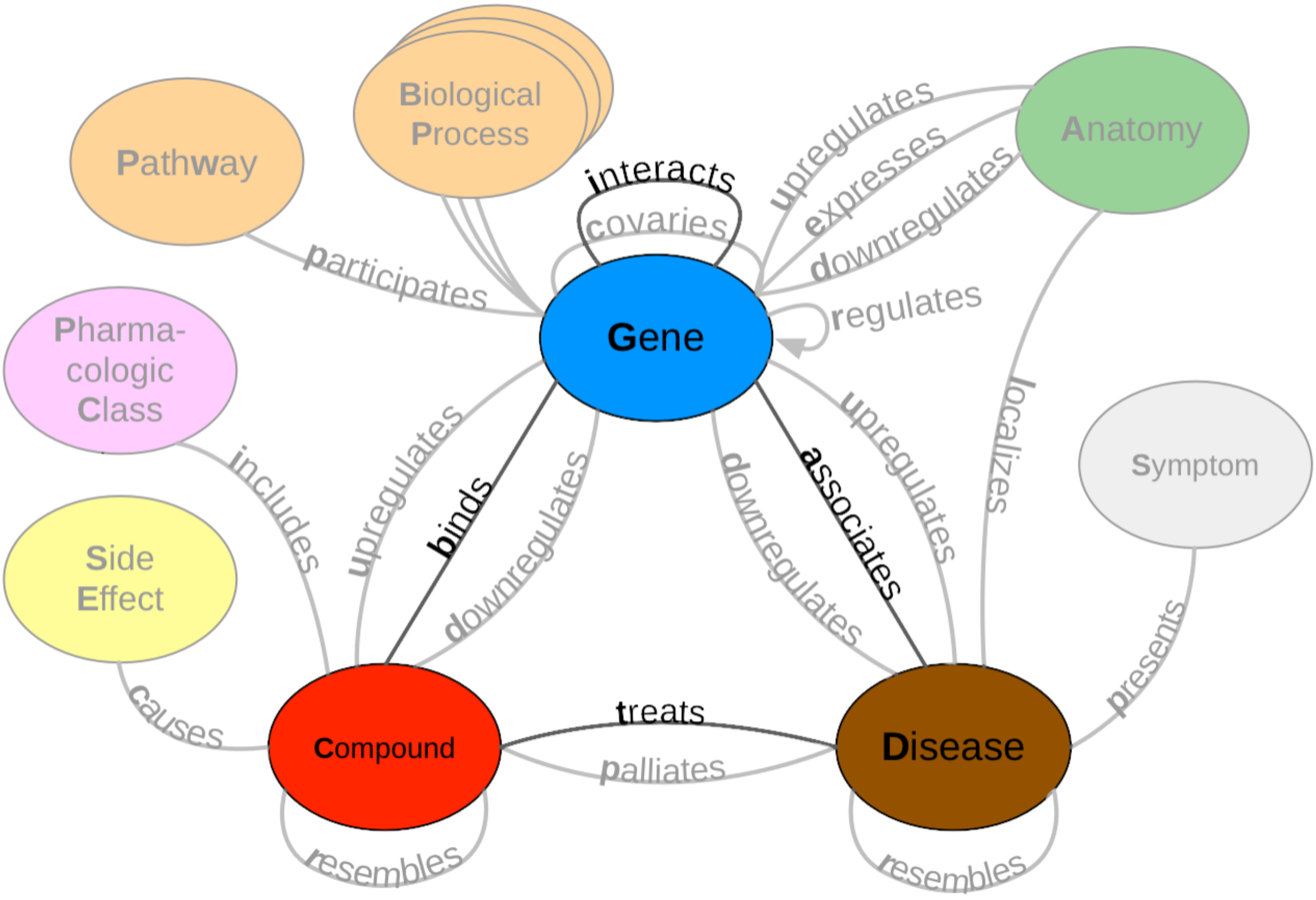

Figure 1: A metagraph (schema) of Hetionet where pharmacological, biological and disease entities are represented as nodes and the relationships between them are represented as edges. This project only focuses on the information shown in bold; however, we can extend this work to incorporate the faded out information as well.

Dataset

We used PubTator [3] as input to our analysis. PubTator provides MEDLINE abstracts that have been annotated with well-established entity recognition tools including DNorm [4] for disease mentions, GeneTUKit [5] for gene mentions, Gnorm [6] for gene normalizations and a dictionary based look system for compound mentions [7]. We downloaded PubTator on June 30, 2017, at which point it contained 10,775,748 abstracts. Then we filtered out mention tags that were not contained in hetionet. We used the Stanford CoreNLP parser [8] to tag parts of speech and generate dependency trees. We extracted sentences with two or more mentions, termed candidate sentences. Each candidates sentence was stratified by co-mention pair to produce a training set, tuning set and a testing set (shown in Table 1). Each unique co-mention pair is sorted into four categories: (1) in hetionet and has sentences, (2) in hetionet and doesn't have sentences, (3) not in hetionet and does have sentences and (4) not in hetionet and doesn't have sentences. Within these four categories each pair receives their own individual partition rank (continuous number between 0 and 1). Any rank lower than 0.7 is sorted into training set, while any rank greater than 0.7 and lower than 0.9 is assigned to tuning set. The rest of the pairs with a rank greater than or equal to 0.9 is assigned to the test set. Sentences that contain more than one co-mention pair are treated as multiple individual candidates. We hand labeled five hundred to a thousand candidate sentences of each relationship to obtain to obtain a ground truth set (Table 1, dataset).

Table 1: Statistics of Candidate Sentences. We sorted each candidate sentence into a training, tuning and testing set. Numbers in parentheses show the number of positives and negatives that resulted from the hand-labeling process.

Relationship	Train	Tune	Test
Disease Associates Gene	2.35 M	31K (397+, 603-)	313K (351+, 649-)
Compound Binds Gene	1.7M	468K (37+, 463-)	227k (31+, 469-)
Compound Treats Disease	1.013M	96K (96+, 404-)	32K (112+, 388-)
Gene Interacts Gene	12.6M	1.056M (60+, 440-)	257K (76+, 424-)

Label Functions for Annotating Sentences

A common challenge in natural language processing is having too few ground truth annotations, even when textual data are abundant. Data programming circumvents this issue by quickly annotating large datasets by using multiple noisy signals emitted by label functions [9]. Label functions are simple pythonic functions that emit: a positive label (1), a negative label (-1) or abstain from emitting a label (0). We combine these functions using a generative model to output a single annotation, which is a consensus probability score bounded between 0 (low chance of mentioning a relationship) and 1 (high chance of mentioning a relationship). We used these annotations to train a discriminator model that makes the final classification step. Our label functions fall into three categories: databases, text patterns and domain heuristics. We provide examples for the categories, described below, using the following candidate sentence: "PTK6 may be a novel therapeutic target for pancreatic cancer."

Databases: These label functions incorporate existing databases to generate a signal, as seen in distant supervision [10]. These functions detect if a candidate sentence's co-mention pair is present in a given database. If the pair is present, emit a positive label and abstain otherwise. If the pair isn't present in any existing database, then a separate label function will emit a negative label. We use a separate label function to prevent the label imbalance problem. This problem occurs when candidates, that scarcely appear in databases, are drowned out by negative labels. The multitude of negative labels increases the likelihood of misclassification when training the generative model.

$$\Lambda_{DB}(D, G) = \begin{cases} 1 & (D, G) \in DB \\ 0 & otherwise \end{cases}$$

$$\Lambda_{\neg DB}(D, G) = \begin{cases} -1 & (D, G) \notin DB \\ 0 & otherwise \end{cases}$$

Text Patterns: These label functions are designed to use keywords and sentence context to generate a signal. For example, a label function could focus on the number of words between two mentions or focus on the grammatical structure of a sentence. These functions emit a positive or negative label depending on the situation. In general, those focused on keywords emit positives and those focused on negation emit negatives.

$$\Lambda_{TP}(D, G) = \begin{cases} 1 & ''\ target\ '' \in Candidate\ Sentence \\ 0 & otherwise \end{cases}$$

$$\Lambda_{TP}(D, G) = \begin{cases} -1 & ''\ VB\ '' \notin pos\_tags(Candidate\ Sentence) \\ 0 & otherwise \end{cases}$$

Domain Heuristics: These label functions use the other experiment results to generate a signal. For this category, we used dependency path cluster themes generated by Percha et al [11]. If a candidate sentence's dependency path belongs to a previously generated cluster, then the label function will emit a positive label and abstain otherwise.

$$\Lambda_{DH}(D, G) = \begin{cases} 1 & Candidate\ Sentence \in Cluster\ Theme \\ 0 & otherwise \end{cases}$$

Roughly half of our label functions are based on text patterns, while the others are distributed across the databases and domain heuristics (Table 2).

Table 2: The distribution of each label function per relationship.

Relationship	Databases (DB)	Text Patterns (TP)	Domain Heuristics (DH)
Disease associates Gene (DaG)	7	20	10
Compound treats Disease (CtD)	3	15	7
Compound binds Gene (CbG)	9	13	7
Gene interacts Gene (GiG)	9	20	8

Training Models

Generative Model

The generative model is a core part of this automatic annotation framework. It integrates multiple signals emitted by label functions and assigns a training class to each candidate sentence. This model assigns training classes by estimating the joint probability distribution of the latent true class (Y) and label function signals (Λ), $P(\Lambda, Y)$. Assuming each label function is conditionally independent, the joint distribution is defined as follows:

$$P(\Lambda, Y) = \frac{\exp(\sum_{i=1}^{m} \theta^T F_i(\Lambda, y))}{\sum_{\Lambda'} \sum_{y'} \exp(\sum_{i=1}^{m} \theta^T F_i(\Lambda', y'))}$$

where m is the number of candidate sentences, F is the vector of summary statistics and θ is a vector of weights for each summary statistic. The summary statistics used by the generative model are as follows:

$$F^{Lab}_{i,j}(\Lambda, Y) = \mathbb{1}\{\Lambda_{i,j} \neq 0\}$$
$$F^{Acc}_{i,j}(\Lambda, Y) = \mathbb{1}\{\Lambda_{i,j} = y_{i,j}\}$$

Lab is the label function's propensity (the frequency of a label function emitting a signal). *Acc* is the individual label function's accuracy given the training class. This model optimizes the weights (θ) by minimizing the negative log likelihood:

$$\hat{\theta} = argmin_{\theta} -\sum_{\Lambda} log \sum_{Y} P(\Lambda, Y)$$

In the framework we used predictions from the generative model, $\hat{Y} = P(Y \mid \Lambda)$, as training classes for our dataset [12,13].

Word Embeddings

Word embeddings are representations that map individual words to real valued vectors of user-specified dimensions. These embeddings have been shown to capture the semantic and syntatic information between words [14]. Using all candidate sentences for each individual relationship pair, we trained facebook's fastText [15] to generate word embeddings. The fastText model uses a skipgram model [16] that aims to predict the context given a candidate word and pairs the model with a novel scoring function that treats each word as a bag of character n-grams. We trained this model for 20 epochs using a window size of 2 and generated 300-dimensional word embeddings. We use the optimized word embeddings to train a discriminative model.

Discriminator Model

talk about the discriminator model and how it works ### Discriminator Model Calibration talk about calibrating deep learning models with temperature smoothing

Experimental Design

Being able to re-use label functions across edge types would substantially reduce the number of label functions required to extract multiple relationship types from biomedical literature. We first established a baseline by training a generative model using only distant supervision label functions designed for the target edge type. As an example, for the gene-interacts-gene edge type we used label functions that returned a `1` if the pair of genes were included in the Human Interaction database [17], the iRefIndex database [18] or in the Incomplete Interactome database [19]. Then we compared models that also included text and domain-heuristic label functions. Using a sampling with replacement approach, we sampled these text and domain-heuristic label functions separately within edge types, across edge types, and from a pool of all label functions. We compared within-edge-type performance to across-edge-type and all-edge-type performance. For each edge type we sampled a fixed number of label functions consisting of five evenly-spaced numbers between one and the total number of possible label functions. We repeated this sampling process 50 times for each point. We evaluated both generative and discriminative models at each point, and we report performance of each in terms of the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR).

Results

Random Sampling of Generative Model

place the grid aurocs here for generative model

Discriminator Model Builds Off Generative Model

place the grid of aurocs here for discriminator model

Random Noise Generative Model

place the results of random label function experiment

Reconstructing Hetionet

place figure of number of new edges that can be added to hetionet as well as edges we can reconstruct using this method

Discussion

Here mention why performnace increases in the beginning for the generative model then decreases

Discuss discriminator model performance given generative model

Mention Take home messages

1. have a centralized set of negative label functions and focus more on contstructing positive label functions

Conclusion and Future Direction

Recap the original problem - takes a long time to create useful label function

Proposed solution - reuse label functions

Mention incorporating more relationships Mention creating a centralized multitask text extractor using this method.

References

1. **Systematic integration of biomedical knowledge prioritizes drugs for repurposing**
Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, Sergio E Baranzini
eLife (2017-09-22) https://doi.org/cdfk
DOI: 10.7554/elife.26726 · PMID: 28936969 · PMCID: PMC5640425

2. **The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog)**
Jacqueline MacArthur, Emily Bowler, Maria Cerezo, Laurent Gil, Peggy Hall, Emma Hastings, Heather Junkins, Aoife McMahon, Annalisa Milano, Joannella Morales, … Helen Parkinson
Nucleic Acids Research (2016-11-29) https://doi.org/f9v7cp
DOI: 10.1093/nar/gkw1133 · PMID: 27899670 · PMCID: PMC5210590

3. **PubTator: a web-based text mining tool for assisting biocuration**
Chih-Hsuan Wei, Hung-Yu Kao, Zhiyong Lu
Nucleic Acids Research (2013-05-22) https://doi.org/f475th
DOI: 10.1093/nar/gkt441 · PMID: 23703206 · PMCID: PMC3692066

4. **DNorm: disease name normalization with pairwise learning to rank**
R. Leaman, R. Islamaj Dogan, Z. Lu
Bioinformatics (2013-08-21) https://doi.org/f5gj9n
DOI: 10.1093/bioinformatics/btt474 · PMID: 23969135 · PMCID: PMC3810844

5. **GeneTUKit: a software for document-level gene normalization**
M. Huang, J. Liu, X. Zhu
Bioinformatics (2011-02-08) https://doi.org/dng2cb
DOI: 10.1093/bioinformatics/btr042 · PMID: 21303863 · PMCID: PMC3065680

6. **Cross-species gene normalization by species inference**
Chih-Hsuan Wei, Hung-Yu Kao
BMC Bioinformatics (2011-10-03) https://doi.org/dnmvds
DOI: 10.1186/1471-2105-12-s8-s5 · PMID: 22151999 · PMCID: PMC3269940

7. **Collaborative biocuration–text-mining development task for document prioritization for curation**
T. C. Wiegers, A. P. Davis, C. J. Mattingly
Database (2012-11-22) https://doi.org/gbb3zw
DOI: 10.1093/database/bas037 · PMID: 23180769 · PMCID: PMC3504477

8. **The Stanford CoreNLP Natural Language Processing Toolkit**
Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, David McClosky
Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2014) https://doi.org/gf3xhp
DOI: 10.3115/v1/p14-5010

9. **Data Programming: Creating Large Training Sets, Quickly**
Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, Christopher Ré
arXiv (2016-05-25) https://arxiv.org/abs/1605.07723v3

10. **Distant supervision for relation extraction without labeled data**
Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky

Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09 (2009) https://doi.org/fg9q43
DOI: 10.3115/1690219.1690287

11. **A global network of biomedical relationships derived from text**
Bethany Percha, Russ B Altman
Bioinformatics (2018-02-27) https://doi.org/gc3ndk
DOI: 10.1093/bioinformatics/bty114 · PMID: 29490008 · PMCID: PMC6061699

12. **Snorkel MeTaL**
Alex Ratner, Braden Hancock, Jared Dunnmon, Roger Goldman, Christopher Ré
Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning - DEEM'18 (2018) https://doi.org/gf3xk7
DOI: 10.1145/3209889.3209898 · PMID: 30931438 · PMCID: PMC6436830

13. **Snorkel**
Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, Christopher Ré
Proceedings of the VLDB Endowment (2017-11-01) https://doi.org/ch44
DOI: 10.14778/3157794.3157797 · PMID: 29770249 · PMCID: PMC5951191

14. **Distributed Representations of Words and Phrases and their Compositionality**
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean
arXiv (2013-10-16) https://arxiv.org/abs/1310.4546v1

15. **Enriching Word Vectors with Subword Information**
Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov
arXiv (2016-07-15) https://arxiv.org/abs/1607.04606v2

16. **Efficient Estimation of Word Representations in Vector Space**
Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean
arXiv (2013-01-16) https://arxiv.org/abs/1301.3781v3

17. **A Proteome-Scale Map of the Human Interactome Network**
Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, … Marc Vidal
Cell (2014-11) https://doi.org/f3mn6x
DOI: 10.1016/j.cell.2014.10.050 · PMID: 25416956 · PMCID: PMC4266588

18. **iRefIndex: A consolidated protein interaction database with provenance**
Sabry Razick, George Magklaras, Ian M Donaldson
BMC Bioinformatics (2008) https://doi.org/b99bjj
DOI: 10.1186/1471-2105-9-405 · PMID: 18823568 · PMCID: PMC2573892

19. **Uncovering disease-disease relationships through the incomplete interactome**
J. Menche, A. Sharma, M. Kitsak, S. D. Ghiassian, M. Vidal, J. Loscalzo, A.-L. Barabasi
Science (2015-02-19) https://doi.org/f3mn6z
DOI: 10.1126/science.1257601 · PMID: 25700523 · PMCID: PMC4435741