Mining Heterogenous Relationships from Pubmed Abstracts Using Weak Supervision

This manuscript (permalink) was automatically generated from greenelab/text_mined_hetnet_manuscript@af32497 on July 24, 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

Knowledge bases are important resources that hold complex structured and unstructed information. These resources have been used in important tasks such as network analysis for drug repurposing discovery [1,2,3] or as a source of training labels for text mining systems [4,5,6]. Populating knowledge bases often requires highly-trained scientists to read biomedical literature and summarize the results [7]. This manual curation process requires a significant amount of effort and time: in 2007 researchers estimated that filling in the missing annotations at that point would require approximately 8.4 years [8]). The rate of publications has continued to increase exponentially [9]. This has been recognized as a considerable challenge and leads to gaps in knowledge bases [8].
Relationship extraction has been studied as a solution towards handling this problem [7]. This process consists of creating a machine learning system to automatically scan and extract relationships from textual sources. Machine learning methods often leverage a large corpus of well-labeled training data, which still requires manual curation. Distant supervision is one technique to sidestep the requirement of well-annotated sentences: with distant supervision one makes the assumption that that all sentences containing an entity pair found in a selected database provide evidence for a relationship [4]. Distant supervision provides many labeled examples; however it is accompanied by a decrease in the quality of the labels.
Ratner et al. [10] recently introduced “data programming” as a solution. Data programming combines distant supervision with the automated labeling of text using hand-written label functions. The distant supervision sources and label functions are integrated using a noise aware generative model, which is used to produce training labels. Combining distant supervision with label functions can dramatically reduce the time required to acquire sufficient training data. However, constructing a knowledge base of heterogeneous relationships through this framework still requires tens of hand-written label functions for each relationship type. Writing useful label functions requires significant error analysis, which can be a time-consuming process.

In this paper, we aim to address the question: to what extent can label functions be re-used across different relationship types? We hypothesized that sentences describing one relationship type may share information in the form of keywords or sentence structure with sentences that indicate other relationship types. We designed a series of experiments to determine the extent to which label function re-use enhanced performance over distant supervision alone. We examine relationships that indicate similar types of physical interactions (i.e., gene-binds-gene and compound-binds-gene) as well as different types (i.e., disease-associates-gene and compound-treats-disease). The re-use of label functions could dramatically reduce the number required to generate and update a heterogeneous knowledge graph.

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 [3] 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 [11] and DrugBank [12]. 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).

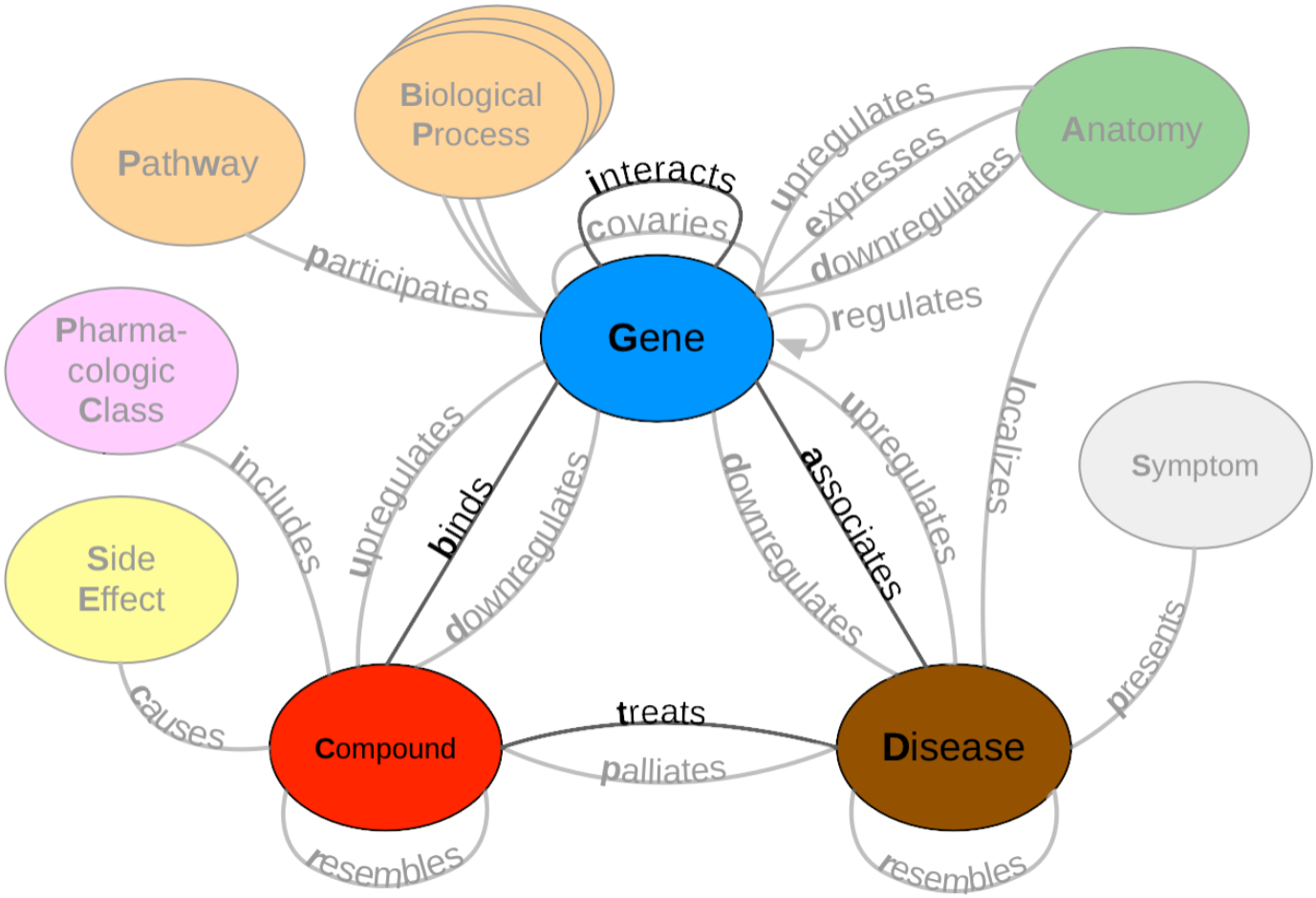

Figure 1: A metagraph (schema) of Hetionet where pharmacological, biological and disease entities are represented as nodes and the relationships between them are represented as edges. This project only focuses on the information shown in bold; however, we can extend this work to incorporate the faded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

Dataset

We used PubTator [13] as input to our analysis. PubTator provides MEDLINE abstracts that have been annotated with well-established entity recognition tools including DNorm [14] for disease mentions, GeneTUKit [15] for gene mentions, Gnorm [16] for gene normalizations and a dictionary based look system for compound mentions [17]. 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 [18] 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 [10]. 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 [4]. 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.

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

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

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 [19]. 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:

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

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 [20,21].

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 [22]. Using all candidate sentences for each individual relationship pair, we trained facebook's fastText [23] to generate word embeddings. The fastText model uses a skipgram model [24] 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.

Discriminative Model

The discriminative model is a neural network, which we train to predict labels from the generative model. The expectation is that the discriminative model can learn more complete features of the text than the label functions that are used in the generative model. We used a convolutional neural network with multiple filters as our discriminative model. This network uses multiple filters with fixed widths of 300 dimensions and a fixed height of 7 (Figure 2), because this height provided the best performance in terms of relationship classification [25]. We trained this model for 20 epochs using the adam optimizer [26] with a learning rate of 0.001. This optimizer used pytorch's default parameter settings. We added a L2 penalty on the network weights to prevent overfitting. Lastly, we added a dropout layer (p=0.25) between the fully connected layer and the softmax layer.

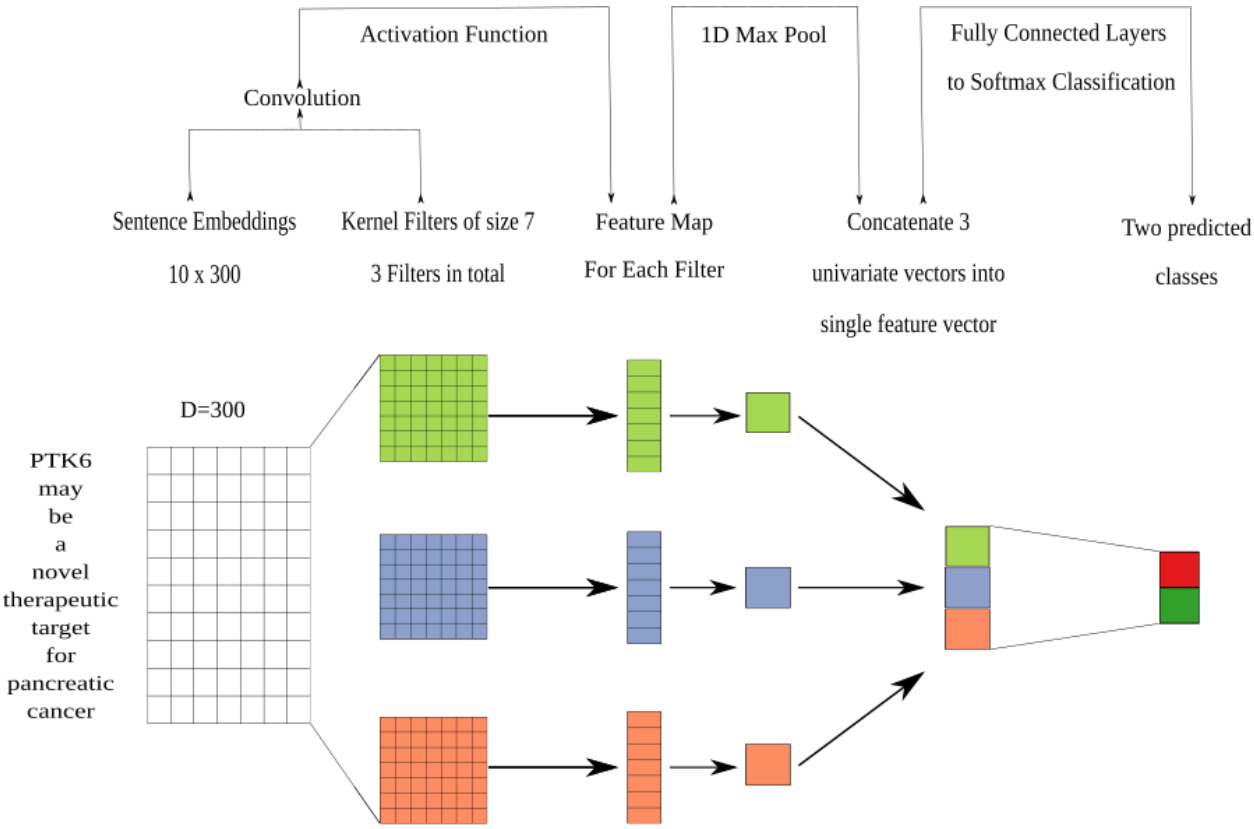

Figure 2: The architecture of the discriminative model is a convolutional neural network. We perform a convolution step using multiple filters. These filters generate a feature map that is sent into a maximum pooling layer. This layer extracts the largest feature in each of these maps. The extracted features are concatenated into a singular vector that is passed into a fully connected network. The fully connected network has 300 neurons for the first layer, 100 neurons for the second layer and 50 neurons for the last layer. From the fully connected network the last step is to generate predictions using the softmax layer.

Calibration of the Discriminative Model

Often many tasks require a machine learning model to output reliable probability predictions. A model is well calibrated if the probabilities emitted from the model match the observed probabilities: a well-calibrated model that assigns a class label with 80% probability should have that class appear 80% of the time. Deep neural network models can often be poorly calibrated [27,28]. These models are usually over-confidenent in their predictions. As a result, we calibrated our convolutional neural network using temperature scaling. Temperature scaling uses a parameter T to scale each value of the logit vector (z) before being passed into the softmax (SM) function.

$$\sigma_{SM}(\frac{z_i}{T}) = \frac{\exp(\frac{z_i}{T})}{\sum_i \exp(\frac{z_i}{T})}$$

We found the optimial T by minimizing the negative log likelihood (NLL) of a held out validation set. The benefit of using this method is the model becomes more reliable and the accuracy of the model doesn't change [27].

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 [29], the iRefIndex database [30] or in the Incomplete Interactome database [31]. 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

Generative Model Using Randomly Sampled Label Functions

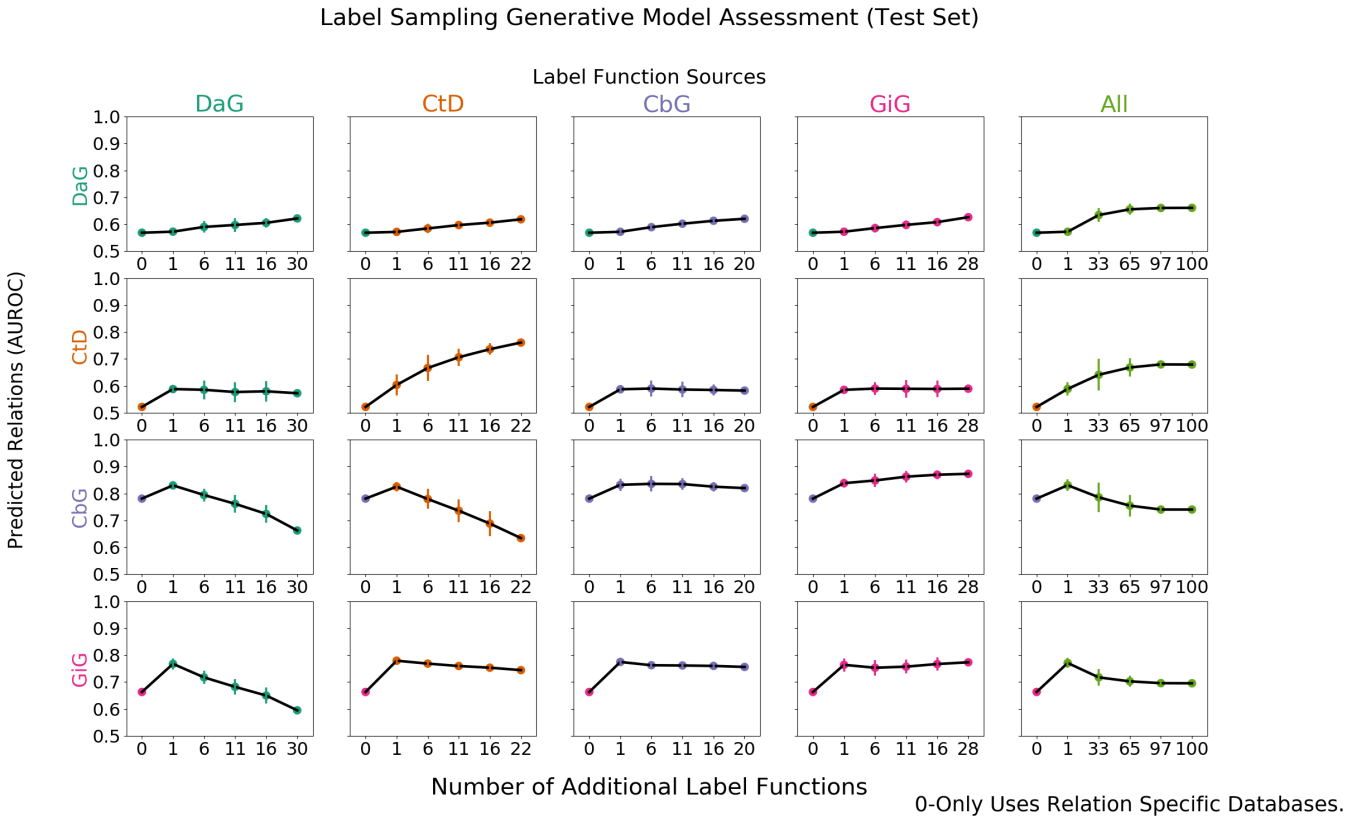

Figure 3: Grid of Area Under the Receiver Operating Curve (AUROC) scores for each generative model trained on randomly sampled label functions. The rows depict the relationship each model is trying to predict and the columns are the relationship specific sources each label function is sampled from. For example, the top-left most square depicts the generative model predicting Disease associates Gene (DaG) sentences, while randomly sampling label functions designed to predict the DaG relationship. The square towards the right depicts the generative model predicting DaG sentences, while randomly sampling label functions designed to predict the Compound treats Disease (CtD) relationship. This pattern continues filling out the rest of the grid. The last most column consists of pooling every relationship specific label function and proceeding as above.

We added randomly sampled label functions to a baseline for each edge type to evaluate the feasibility of label function re-use. Our baseline model consisted of a generative model trained with only the edge type's distant supervision label functions. We report the results in the form of area under the precision recall curve (AUPR) (Figure 4) and area under the receiver operating curve (AUROC) (Figure 3).
The on-diagonal plots of figure 4) and figure 4 show performance when edge-specific label functions are added on top of edge-specific baselines. The general trend is performance increases in this setting. The Compound-treats-Disease (CtD) edge type is a quintessential example of this trend. The baseline model starts off with an AUROC score of 52% and an AUPRC of 28%, which increase to 76% and 49% respectively as more CtD label functions are included. Disease-associates-Gene (DaG) edges have a similar trend: performance starting off with a AUROC of 56% and AUPRC of 41%, which increase to 62% and 45% respectively. Both the Compound-binds-Gene (CbG) and Gene-interacts-Gene (GiG) edges have an increasing trend but plateau after a few label functions are added.

The off-diagonals in figure 4) and figure 4 show how performance varies when label functions from one edge type are added to a different edge type's baseline. In certain cases (apparent for DaG), performance increases regardless of the edge type used for label functions. In other cases (apparent with CtD), one label function appears to improve performance; however, adding more label functions does not improve performance (AUROC) or decreases it (AUPRC). In certain cases, the source of the label functions appear to be important: for CbG edges performance decreases when using label functions from the DaG and CtD categories.

Our initial hypothesis was based on the idea that certain edge types capture similar physical relationships and that these cases would be particularly amenable for label function transfer. For example, Compound-binds-Gene (CbG) and Gene-interacts-Gene (GiG) both describe physical interactions. We observed that performance increased as assessed by both AUPRC and AUPRC when using label functions from the GiG edge type to predict CbG edges. A similar trend was observed when predicting the GiG edge; however, the performance differences were small for this edge type making the importance difficult to assess.
The last column shows performance when sampling from all label functions. Performance increased (AUROC and AUPRC) for both DaG and CtD, when sampling from the full pool of label functions. CbG and GiG also had increased performance when one random label function was sampled, but performance decreased drastically as more label functions were added. It is possible that a small number of irrelevant label functions are able to overwhelm the distant supervision label functions in these cases.

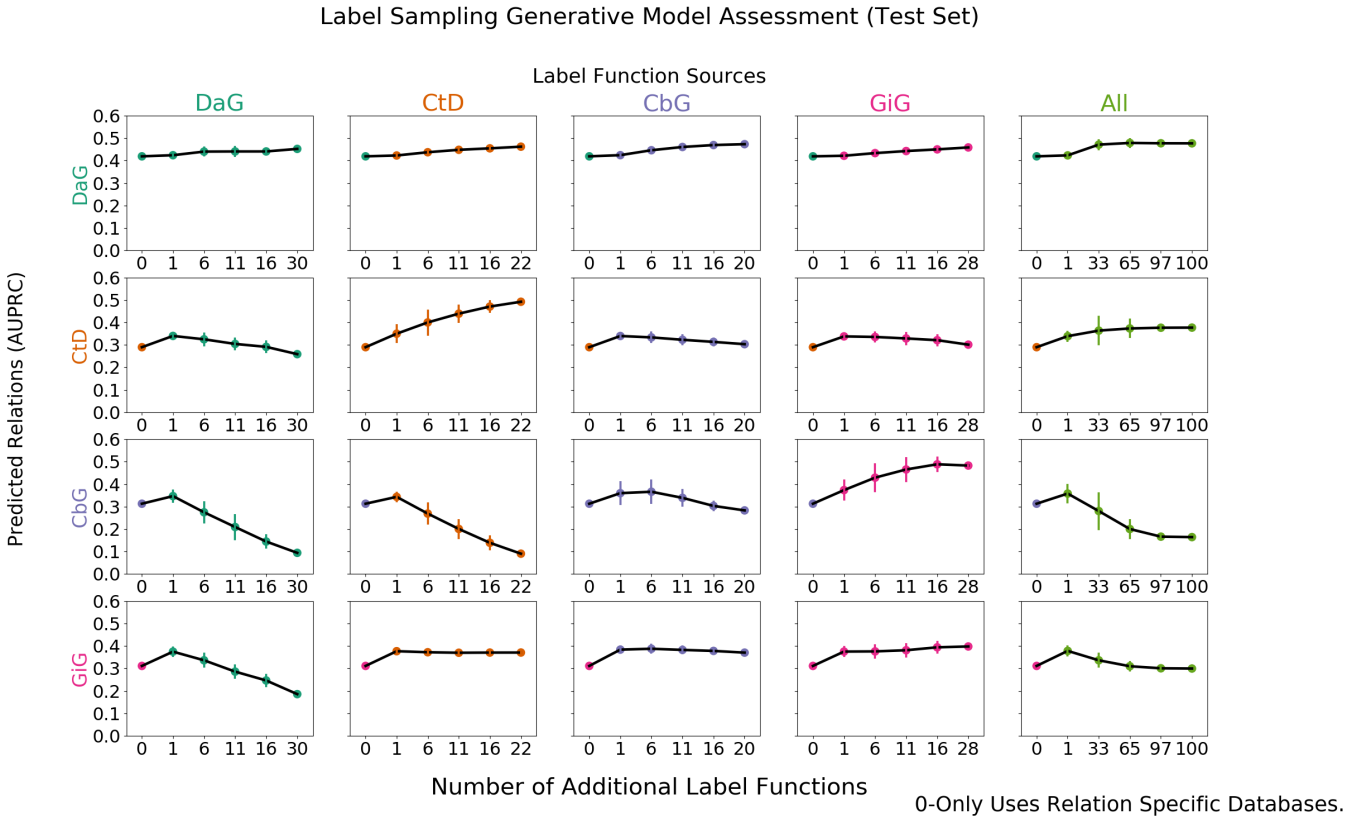

Figure 4: Grid of Area Under the Precision Recall Curve (AUPRC) scores for each generative model trained on randomly sampled label functions. The rows depict the relationship each model is trying to predict and the columns are the relationship specific sources each label function is sampled from. For example, the top-left most square depicts the generative model predicting Disease associates Gene (DaG) sentences, while randomly sampling label functions designed to predict the DaG relationship. The square towards the right depicts the generative model predicting DaG sentences, while randomly sampling label functions designed to predict the Compound treats Disease (CtD) relationship. This pattern continues filling out the rest of the grid. The last most column consists of pooling every relationship specific label function and proceeding as above.

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

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. **Graph Theory Enables Drug Repurposing – How a Mathematical Model Can Drive the Discovery of Hidden Mechanisms of Action**
Ruggero Gramatica, T. Di Matteo, Stefano Giorgetti, Massimo Barbiani, Dorian Bevec, Tomaso Aste
PLoS ONE (2014-01-09) https://doi.org/gf45zp
DOI: 10.1371/journal.pone.0084912 · PMID: 24416311 · PMCID: PMC3886994

2. **Drug repurposing through joint learning on knowledge graphs and literature**
Mona Alshahrani, Robert Hoehndorf
Cold Spring Harbor Laboratory (2018-08-06) https://doi.org/gf45zk
DOI: 10.1101/385617

3. **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

4. **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

5. **CoCoScore: Context-aware co-occurrence scoring for text mining applications using distant supervision**
Alexander Junge, Lars Juhl Jensen
Cold Spring Harbor Laboratory (2018-10-16) https://doi.org/gf45zm
DOI: 10.1101/444398

6. **Knowledge-guided convolutional networks for chemical-disease relation extraction**
Huiwei Zhou, Chengkun Lang, Zhuang Liu, Shixian Ning, Yingyu Lin, Lei Du
BMC Bioinformatics (2019-05-21) https://doi.org/gf45zn
DOI: 10.1186/s12859-019-2873-7 · PMID: 31113357 · PMCID: PMC6528333

7. **Facts from text: can text mining help to scale-up high-quality manual curation of gene products with ontologies?**
R. Winnenburg, T. Wachter, C. Plake, A. Doms, M. Schroeder
Briefings in Bioinformatics (2008-07-11) https://doi.org/bfsnwg
DOI: 10.1093/bib/bbn043 · PMID: 19060303

8. **Manual curation is not sufficient for annotation of genomic databases**
William A. Baumgartner Jr, K. Bretonnel Cohen, Lynne M. Fox, George Acquaah-Mensah, Lawrence Hunter
Bioinformatics (2007-07-01) https://doi.org/dtck86
DOI: 10.1093/bioinformatics/btm229 · PMID: 17646325 · PMCID: PMC2516305

9. **Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references**
Lutz Bornmann, Rüdiger Mutz

Journal of the Association for Information Science and Technology (2015-04-29) https://doi.org/gfj5zc
DOI: 10.1002/asi.23329

10. **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

11. **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

12. **DrugBank 5.0: a major update to the DrugBank database for 2018**
David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, … Michael Wilson
Nucleic Acids Research (2017-11-08) https://doi.org/gcwtzk
DOI: 10.1093/nar/gkx1037 · PMID: 29126136 · PMCID: PMC5753335

13. **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

14. **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

15. **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

16. **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

17. **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

18. **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

19. **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

20. **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

21. **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

22. **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

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

24. **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

25. **A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification**
Ye Zhang, Byron Wallace
arXiv (2015-10-13) https://arxiv.org/abs/1510.03820v4

26. **Adam: A Method for Stochastic Optimization**
Diederik P. Kingma, Jimmy Ba
arXiv (2014-12-22) https://arxiv.org/abs/1412.6980v9

27. **On Calibration of Modern Neural Networks**
Chuan Guo, Geoff Pleiss, Yu Sun, Kilian Q. Weinberger
arXiv (2017-06-14) https://arxiv.org/abs/1706.04599v2

28. **Accurate Uncertainties for Deep Learning Using Calibrated Regression**
Volodymyr Kuleshov, Nathan Fenner, Stefano Ermon
arXiv (2018-07-01) https://arxiv.org/abs/1807.00263v1

29. **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

30. **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

31. **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