Expanding a Database-derived Biomedical Knowledge Graph via Multi-relation Extraction from Biomedical Abstracts

David N. Nicholson[1], Daniel S. Himmelstein[1] and Casey S. Greene[1]

[1]*Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, 3400 Civic Center Blvd, Philadelphia, PA 19104*

Abstract

Knowledge graphs support multiple research efforts by providing contextual information for biomedical entities, constructing networks, and supporting the interpretation of high-throughput analyses. These databases are populated via some form of manual curation, which is difficult to scale in the context of an increasing publication rate. Data programming is a paradigm that circumvents this arduous manual process by combining databases with simple rules and heuristics written as label functions, which are programs designed to automatically annotate textual data. Unfortunately, writing a useful label function requires substantial error analysis and is a nontrivial task that takes multiple days per function. This makes populating a knowledge graph with multiple nodes and edge types practically infeasible. We sought to accelerate the label function creation process by evaluating the extent to which label functions could be re-used across multiple edge types. We used a subset of an existing knowledge graph centered on disease, compound, and gene entities to evaluate label function re-use. We determined the best label function combination by comparing a baseline database-only model with the same model but added edge-specific or edge-mismatch label functions. We confirmed that adding additional edge-specific rather than edge-mismatch label functions often improves text annotation and shows that this approach can incorporate novel edges into our source knowledge graph. We expect that continued development of this strategy has the potential to swiftly populate knowledge graphs with new discoveries, ensuring that these resources include cutting-edge results.

Introduction

Knowledge bases are important resources that hold complex structured and unstructured 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 time-consuming process is referred to as manual curation. In 2007, researchers estimated that filling a knowledge base via manual curation would require approximately 8.4 years to complete [8]. The rate of publications continues to exponentially increase [9], so using only manual curation to fully populate a knowledge base has become impractical.

Relationship extraction has been studied as a solution towards handling the challenge posed by an exponentially growing body of literature [7]. This process consists of creating an expert system to automatically scan, detect and extract relationships from textual sources. Typically, these systems utilize machine learning techniques that require extensive corpora of well-labeled

training data. These corpora are difficult to obtain, because they are constructed via extensive manual curation pipelines.

Distant supervision is a technique also designed to sidestep the dependence on manual curation and quickly generate large training datasets. This technique assumes that positive examples established in selected databases can be applied to any sentence that contains them [4]. The central problem with this technique is that generated labels are often of low quality which results in an expansive amount of false positives [10].

Ratner et al. [11] recently introduced "data programming" as a solution. Data programming is a paradigm that combines distant supervision with simple rules and heuristics written as small programs called label functions. These label functions are consolidated via a noise aware generative model that is designed to produce training labels for large datasets. Using this paradigm can dramatically reduce the time required to obtain sufficient training data; however, writing a useful label function requires a significant amount of time and error analysis. This dependency makes constructing a knowledge base with a myriad of heterogenous relationships nearly impossible as tens or possibly hundreds of label functions are required per relationship type.

In this paper, we seek to accelerate the label function creation process by measuring the extent to which label functions can be re-used across different relationship types. We hypothesize that sentences describing one relationship type may share linguistic features such as keywords or sentence structure with sentences describing other relationship types. We conducted a series of experiments to determine the degree to which label function re-use enhanced performance over distant supervision alone. We focus on 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). Re-using label functions could dramatically reduce the time required to populate a knowledge base with a multitude of heterogeneous relationships.

Related Work

Relationship extraction is the process of detecting semantic relationships from a collection of text. This process can be broken down into three different categories: (1) the use of natural language processing techniques such as manually crafted rules and heuristics for relationship extraction (Rule Based Extractors), (2) the use of unsupervised methods such as co-occurrence scores or clustering to find patterns within sentences and documents (Unsupervised Extractors), and (3) the use of supervised or semi-supervised machine learning for classifying the presence of a relation within documents or sentences (Supervised Extractors). In this section, we briefly discuss selected efforts under each category.

Rule Based Extractors

Rule based extractors rely heavily on expert knowledge to perform extraction. Typically, these systems use linguistic rules and heuristics to identify key sentences or phrases. For example, a hypothetical extractor focused on protein phosphorylation events would identify sentences containing the phrase "gene X phosphorylates gene Y" [12]. This phrase is a straightforward indication that two genes have a fundamental role in protein phosphorylation. Other phrase extractors have been used to identify drug-disease treatments [13], pharmcogenomic events [14]

and protein-protein interactions [15,16]. These extractors provide a simple and effective way to extract sentences; however, they depend on extensive knowledge about the text to be properly constructed.

A sentence's grammatical structure can also support relationship extraction via dependency trees. Dependency trees are data structures that depict a sentence's grammatical relation structure in the form of nodes and edges. Nodes represent words and edges represent the dependency type each word shares between one another. For example, a possible extractor would classify sentences as a positive if a sentence contained the following dependency tree path: "gene X (subject)-> promotes (verb)<- cell death (direct object) <- in (preposition) <-tumors (object of preposition)" [17]. This approach provides extremely precise results, but the quantity of positive results remains modest as sentences appear in distinct forms and structure. Because of this limitation, recent approaches have incorporated methods on top of rule based extractors such as co-occurrence and machine learning systems [18,19]. We discuss the pros and cons of added methods in a later section. For this project, we constructed our label functions without the aid of these works; however, approaches discussed in this section provide substantial inspiration for novel label functions in future endeavors.

Unsupervised Extractors

Unsupervised extractors detect relationships without the need of annotated text. Notable approaches exploit the fact that two entities can occur together in text. This event is referred to as co-occurrence. Extractors utilize these events by generating statistics on the frequency of entity pairs occurring in text. For example, a possible extractor would say gene X is associated with disease Y, because gene X and disease Y appear together more often than individually [20]. This approach has been used to establish the following relationship types: disease-gene relationships [20,21,22,23,24,25], protein-protein interactions [24,26,27], drug-disease treatments [28], and tissue-gene relations [29]. Extractors using the co-occurrence strategy provide exceptional recall results; however, these methods may fail to detect underreported relationships, because they depend on entity-pair frequency for detection. Junge et al. created a hybrid approach to account for this issue using distant supervision to train a classifier to learn the context of each sentence [30]. Once the classifier was trained, they scored every sentence within their corpus, and each sentence's score was incorporated into calculating co-occurrence frequencies to establish relationship existence [30]. Co-occurrence approaches are powerful in establishing edges on the global scale; however, they cannot identify individual sentences without the need for supervised methods.

Clustering is an unsupervised approach that extracts relationships from text by grouping similar sentences together. Percha et al. used this technique to group sentences based on their grammatical structure [31]. Using Stanford's Core NLP Parser [32], a dependency tree was generated for every sentence in each Pubmed abstract [31]. Each tree was clustered based on similarity and each cluster was manually annotated to determine which relationship each group represented [31]. For our project we incorporated the results of this work as domain heuristic label functions. Overall, unsupervised approaches are desirable since they do not require well-annotated training data. Such approaches provide excellent recall; however, performance can be limited in terms of precision when compared to supervised machine learning methods [33,34].

Supervised Extractors

Supervised extractors consist of training a machine learning classifier to predict the existence of a relationship within text. These classifiers require access to well-annotated datasets, which are usually created via some form of manual curation. Previous work consists of research experts curating their own datasets to train classifiers [35,36,37,38,39]; however, there have been community-wide efforts to create datasets for shared tasks [40,41,42]. Shared tasks are open challenges that aim to build the best classifier for natural language processing tasks such as named entity tagging or relationship extraction. A notable example is the BioCreative community that hosted a number of shared tasks such as predicting compound-protein interactions (BioCreative VI track 5) [41] and compound induced diseases [42]. Often these datasets are well annotated, but are modest in size (2,432 abstracts for BioCreative VI [41] and 1500 abstracts for BioCreative V [42]). As machine learning classifiers become increasingly complex, these small dataset sizes cannot suffice. Plus, these multitude of datasets are uniquely annotated which can generate noticeable differences in terms of classifier performance [42]. Overall, obtaining large well-annotated datasets still remains as an open non-trivial task.

Before the rise of deep learning, a classifier that was most frequently used was support vector machines. This classifier uses a projection function called a kernel to map data onto a high dimensional space so datapoints can be easily discerned between classes [43]. This method was used to extract disease-gene associations [35,44,45], protein-protein interactions[19,46,47] and protein docking information [48]. Generally, support vector machines perform well on small datasets with large feature spaces but are slow to train as the number of datapoints becomes asymptotically large.

Deep learning has been increasingly popular as these methods can outperform common machine learning methods [49]. Approaches in this field consist of using various neural network architectures, such as recurrent neural networks [50,51,52,53,54,55] and convolutional neural networks [51,54,56,57,58], to extract relationships from text. In fact approaches in this field were the winning model within the BioCreative VI shared task [41,59]. Despite the substantial success of these models, they often require large amounts of data to perform well. Obtaining large datasets is a time-consuming task, which makes training these models a non-trivial challenge. Distant supervision has been used as a solution to fix the barren amount of large datasets [4]. Approaches have used this paradigm to extract chemical-gene interactions [54], disease-gene associations [30] and protein-protein interactions [30,54,60]. In fact, efforts done in [60] served as one of the motivating rationales for our work.

Overall, deep learning has provided exceptional results in terms of relationships extraction. Thus, we decided to use a deep neural network as our discriminative model.

Methods and Materials

Hetionet

Hetionet v1 [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 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

v1 were obtained from open databases, such as the GWAS Catalog [61] and DrugBank [62]. For this project, we analyzed performance over a subset of the Hetionet v1 edge types: disease associates with a gene (DaG), compound binds to a gene (CbG), compound treating a disease (CtD) and gene interacts with gene (GiG) (bolded in Figure 1).

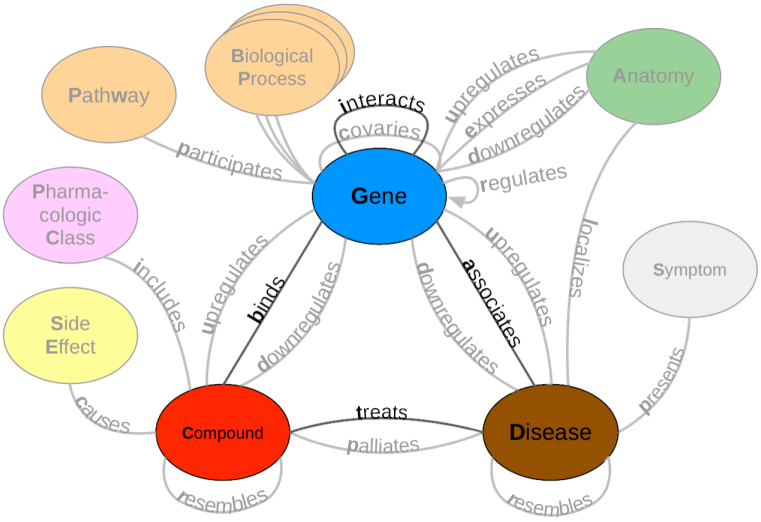

Figure 1: A metagraph (schema) of Hetionet v1 where biomedical entities are represented as nodes and the relationships between them are represented as edges. We examined performance on the highlighted subgraph; however, the long-term vision is to capture edges for the entire graph.

Dataset

We used PubTator [63] as input to our analysis. PubTator provides MEDLINE abstracts that have been annotated with well-established entity recognition tools including DNorm [64] for disease mentions, GeneTUKit [65] for gene mentions, Gnorm [66] for gene normalizations and a dictionary based search system for compound mentions [67]. 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 v1. We used the Stanford CoreNLP parser [32] to tag parts of speech and generate dependency trees. We extracted sentences with two or more mentions, termed candidate sentences. Each candidate sentence was stratified by co-mention pair to produce a training set, tuning set and a testing set (shown in Supplemental Table 2). Each unique co-mention pair was sorted into four categories: (1) in Hetionet v1 and has sentences, (2) in Hetionet v1 and doesn't have sentences, (3) not in Hetionet v1 and does have sentences and (4) not in Hetionet v1 and doesn't have sentences. Within these four categories each pair is randomly assigned their own individual partition rank (a continuous number between 0 and 1). Any rank lower than 0.7 is sorted into the training set, while any rank greater than 0.7 and lower than 0.9 is assigned to the 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 edge type to obtain a ground truth set (Supplemental Table 2)[1].

[1] Labeled sentences are available here.

Label Functions for Annotating Sentences

The challenge of having too few ground truth annotations is common to many natural language processing settings, even when unannotated text is abundant. Data programming circumvents this issue by quickly annotating large datasets by using multiple noisy signals emitted by label functions [11]. Label functions are simple pythonic functions that emit: a positive label (1), a negative label (-1) or abstain from emitting a label (0). These functions can be grouped into multiple categories (see Supplement Methods). We combined 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 discriminative model that makes the final classification step.

Experimental Design

Being able to re-use label functions across edge types would substantially reduce the number of label functions required to extract multiple relationships 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 (see Supplemental Methods). For example, in the Gene interacts Gene (GiG) edge type we used label functions that returned a 1 if the pair of genes were included in the Human Interaction database [68], the iRefIndex database [69] or in the Incomplete Interactome database [70]. Then we compared the baseline model with 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. Furthermore, at each point we also trained the discriminative model using annotations from the generative model trained on edge-specific label functions (see Supplemental Methods). We report performance of both models in terms of the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR). Ensuing model evaluations, we quantified the number of edges we could incorporate into Hetionet v1. Using a calibrated discriminative model (see Supplemental Methods), we scored every candidate sentence within our dataset and grouped candidates based on their mention pair. We took the max score within each candidate group and this score represents the probability of the existence of an edge. We established edges by using a cutoff score that produced an equal error rate between the false positives and false negatives. We report the number of preexisting edges we could recall as well as the number of novel edges we can incorporate. Lastly, we compared our framework with a previously established unsupervised approach [30].

Results

Generative Model Using Randomly Sampled Label Functions

Creating label functions is a labor-intensive process that can take days to accomplish. We sought to accelerate this process by measuring the extent to which label functions can be reused. Our hypothesis was that certain edge types share similar linguistic features such as keywords and/or sentence structure. This shared characteristic would make certain edge types amenable to label

function reuse. We designed a set of experiments to test this hypothesis on an individual level (edge vs edge) as well as a global level (collective pool of sources). We observed that performance increased when edge-specific label functions were added to an edge-specific baseline model, while label function reuse usually provided less benefit (AUROC Figure 2, AUPR Supplemental Figure 5). We also evaluated randomly selecting label functions from among all sets and observed similar performance (AUROC Supplemental Figure 6, AUPR Supplemental Figure 7) The quintessential example of this overarching trend is the Compound treats Disease (CtD) edge type, where edge-specific label functions always outperformed transferred label functions. However, there are hints of label function transferability for selected edge types and label function sources. Performance increases as more CbG label functions are incorporated to the GiG baseline model and vice versa. This suggests that sentences for GiG and CbG may share similar linguistic features or terminology that allows for label functions to be reused. Perplexingly, edge-specific Disease associates Gene (DaG) label functions did not improve performance over label functions drawn from other edge types. Overall, only CbG and GiG showed significant signs of reusability which suggests label functions could be shared between the two edge types.

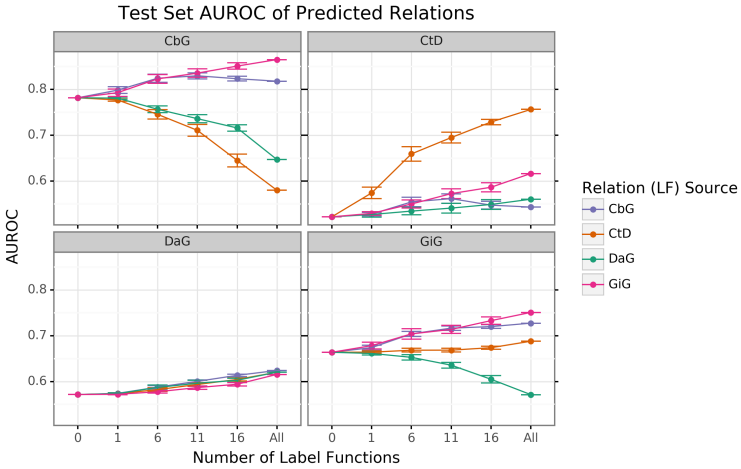

Figure 2: Edge-specific label functions are better performing than edge-mismatch label functions, but certain mismatch situations show signs of successful transfer. Each line plot header depicts the edge type the generative model is trying to predict, while the colors represent the source of label functions. For example, orange represents sampling label functions designed to predict the Compound treats Disease (CtD) edge type. The x axis shows the number of randomly sampled label functions being incorporated into the database-only baseline model (point at 0). The y axis shows area under the receiver operating curve (AUROC). Each point on the plot shows the average of 50 sample runs, while the error bars show the 95% confidence intervals of all runs. The baseline and "All" data points consist of sampling from the entire fixed set of label functions.

We found that sampling from all label function sources at once usually underperformed relative to edge-specific label functions (Supplemental Figures 6 and 7). As more label functions were sampled, the gap between edge-specific sources and all sources widened. CbG is a prime example of this trend (Supplemental Figures 6 and 7), while CtD and GiG show a similar but milder trend. DaG was the exception to the general rule: the pooled set of label functions improved performance over the edge-specific ones, which aligns with the previously observed results for individual edge types (Figure 2). The decreasing trend when pooling all label functions supports the notion that label functions cannot easily transfer between edge types (exception being CbG on GiG and vice versa).

Discriminative Model Performance

The discriminative model is designed to augment performance over the generative model by incorporating textual features along with estimated training labels. The discriminative model is a piecewise convolutional neural network trained over word embeddings (See Methods and Materials). We found that the discriminative model generally out-performed the generative model as more edge-specific label functions are incorporated (Figure 3 and Supplemental Figure 8). The discriminative model's performance is often poorest when very few edge-specific label functions are added to the baseline model (seen in Disease associates Gene (DaG), Compound binds Gene (CbG) and Gene interacts Gene (GiG)). This suggests that generative models trained with more label functions produce outputs that are more suitable for training discriminative models. An exception to this trend is Compound treats Disease (CtD) where the discriminative model out-performs the generative model at all levels of sampling. We observed the opposite trend with the Compound-binds-Gene (CbG) edges: the discriminative model was always poorer or indistinguishable from the generative model. Interestingly, the AUPR for CbG plateaus below the generative model and decreases when all edge-specific label functions are used (Supplemental Figure 8). This suggests that the discriminative model might be predicting more false positives in this setting. Incorporating more edge-specific label functions usually improves performance for the discriminative model over the generative model.

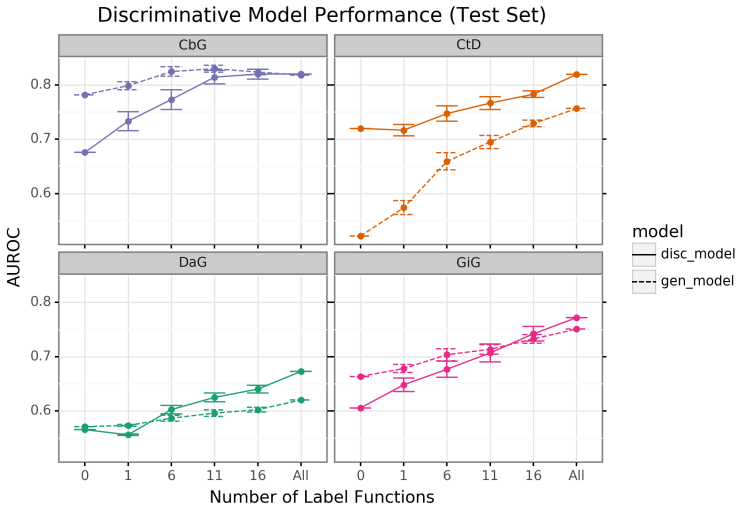

Figure 3: The discriminative model usually improves at a faster rate than the generative model as more edge-specific label function are included. The line plot headers represent the specific edge type the discriminative model is trying to predict. The x-axis shows the number of randomly sampled label functions that are incorporated into the baseline model (point at 0). The y axis shows the area under the receiver operating curve (AUROC). Each datapoint represents the average of 50 sample runs and the error bars represent the 95% confidence interval of each run. The baseline and "All" data points consist of sampling from the entire fixed set of label functions.

Discussion

We measured the extent to which label functions can be re-used across multiple edge types to extract relationships from literature. Through our sampling experiment, we found that adding edge-specific label functions increases performance for the generative model (Figure 2). We found that label functions designed from relatively related edge types can increase performance (Gene interacts Gene (GiG) label functions predicting the Compound binds Gene (CbG) edge

and vice versa), while the Disease associates Gene (DaG) edge type remained agnostic to label function sources (Figure 2 and Supplemental Figure 5). Furthermore, we found that using all label functions at once generally hurts performance with the exception being the DaG edge type (Supplemental Figures 6 and 7). One possibility for this observation is that DaG is a broadly defined edge type. For example, DaG may contain many concepts related to other edge types such as Disease (up/down) regulating a Gene, which makes it more agnostic to label function sources (examples highlighted in our annotated sentences).

Regarding the discriminative model, adding edge-specific label function substantially improved performance for two out of the four edge types (Compound treats Disease (CtD) and Disease associates Gene (DaG)) (Figure 3 and Supplemental Figure 8). Gene interacts Gene (GiG) and Compound binds Gene (CbG) discriminative models showed minor improvements compared to the generative model, but only when nearly all edge-specific label functions are included (Figure 3 and Supplemental Figure 8). We came across a large amount of spurious gene mentions when working with the discriminative model and believe that these mentions contributed to CbG and GiG's hindered performance. We encountered difficulty in calibrating each discriminative model (Supplemental Figure 9). The temperature scaling algorithm appears to improve calibration for the highest scores for each model but did not successfully calibrate throughout the entire range of predictions. Improving performance for all predictions may require more labeled examples or may be a limitation of the approach in this setting. Even with these limitations, this early-stage approach could recall many existing edges from an existing knowledge base, Hetionet v1, and suggest many new high-confidence edges for inclusion (Supplemental Figure 10). Our findings suggest that further work, including an expansion of edge types and a move to full text from abstracts, may make this approach suitable for building continuously updated knowledge bases to address drug repositioning and other biomedical challenges.

Conclusion and Future Direction

Filling out knowledge bases via manual curation can be an arduous and erroneous task [8]. As the rate of publications increases, relying on manual curation alone becomes impractical. Data programming, a paradigm that uses label functions as a means to speed up the annotation process, can be used as a solution for this problem. An obstacle for this paradigm, however, is creating useful label functions, which takes a considerable amount of time. We tested the feasibility of reusing label functions as a way to reduce the total number of label functions required for strong prediction performance. We conclude that label functions may be re-used with closely related edge types, but that re-use does not improve performance for most pairings. The discriminative model's performance improves as more edge-specific label functions are incorporated into the generative model; however, we did notice that performance greatly depends on the annotations provided by the generative model.

This work sets up the foundation for creating a common framework that mines text to create edges. Within this framework we would continuously incorporate new knowledge as novel findings are published, while providing a single confidence score for an edge via sentence score consolidation. As opposed to many existing knowledge graphs (for example, Hetionet v1 where text-derived edges generally cannot be exactly attributed to excerpts from literature [3,71]), our approach has the potential to annotate each edge based on its source sentences. In addition, edges generated with this approach would be unencumbered from upstream licensing or copyright restrictions, enabling openly licensed hetnets at a scale not previously possible [72,73,74]. New

multitask learning [75] strategies may make it even more practical to reuse label functions to construct continuously updating literature-derived knowledge graphs.

Supplemental Information

An online version of this manuscript is available at https://greenelab.github.io/text_mined_hetnet_manuscript/. Source code for this work is available under open licenses at: https://github.com/greenelab/snorkeling/.

Acknowledgements

The authors would like to thank Christopher Ré's group at Stanford University, especially Alex Ratner and Steven Bach, for their assistance with this project. We also want to thank Graciela Gonzalez-Hernandez for her advice and input with this project. This work was support by Grant GBMF4552 from the Gordon Betty Moore Foundation.

References

[1] R. Gramatica, T. Di Matteo, S. Giorgetti, M. Barbiani, D. Bevec, and T. Aste, "Graph Theory Enables Drug Repurposing – How a Mathematical Model Can Drive the Discovery of Hidden Mechanisms of Action," *PLoS ONE*, vol. 9, no. 1, p. e84912, Jan. 2014 [Online]. Available: https://doi.org/gf45zp

[2] M. Alshahrani and R. Hoehndorf, "Drug repurposing through joint learning on knowledge graphs and literature," Cold Spring Harbor Laboratory, 06-Aug-2018 [Online]. Available: https://doi.org/gf45zk

[3] D. S. Himmelstein *et al.*, "Systematic integration of biomedical knowledge prioritizes drugs for repurposing," *eLife*, vol. 6, Sep. 2017 [Online]. Available: https://doi.org/cdfk

[4] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *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 [Online]. Available: https://doi.org/fg9q43

[5] A. Junge and L. J. Jensen, "CoCoScore: Context-aware co-occurrence scoring for text mining applications using distant supervision," Cold Spring Harbor Laboratory, 16-Oct-2018 [Online]. Available: https://doi.org/gf45zm

[6] H. Zhou, C. Lang, Z. Liu, S. Ning, Y. Lin, and L. Du, "Knowledge-guided convolutional networks for chemical-disease relation extraction," *BMC Bioinformatics*, vol. 20, no. 1, May 2019 [Online]. Available: https://doi.org/gf45zn

[7] R. Winnenburg, T. Wachter, C. Plake, A. Doms, and M. Schroeder, "Facts from text: can text mining help to scale-up high-quality manual curation of gene products with ontologies?" *Briefings in Bioinformatics*, vol. 9, no. 6, pp. 466–478, Jul. 2008 [Online]. Available: https://doi.org/bfsnwg

[8] W. A. Baumgartner Jr, K. B. Cohen, L. M. Fox, G. Acquaah-Mensah, and L. Hunter, "Manual curation is not sufficient for annotation of genomic databases," *Bioinformatics*, vol. 23, no. 13, pp. i41–i48, Jul. 2007 [Online]. Available: https://doi.org/dtck86

[9] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *J Assn Inf Sci Tec*, vol. 66, no. 11, pp. 2215–2222, Apr. 2015 [Online]. Available: https://doi.org/gfj5zc

[10] T. Jiang, J. Liu, C.-Y. Lin, and Z. Sui, "Revisiting distant supervision for relation extraction," in *LREC*, 2018.

[11] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré, "Data Programming: Creating Large Training Sets, Quickly," arXiv, 1605.07723v3, May 2016 [Online]. Available: https://arxiv.org/abs/1605.07723v3

[12] M. Torii, C. N. Arighi, G. Li, Q. Wang, C. H. Wu, and K. Vijay-Shanker, "RLIMS-P 2.0: A Generalizable Rule-Based Information Extraction System for Literature Mining of Protein Phosphorylation Information," *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 12, no. 1, pp. 17–29, Jan. 2015 [Online]. Available: https://doi.org/gf8fpv

[13] R. Xu and Q. Wang, "Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing," *BMC Bioinformatics*, vol. 14, no. 1, Jun. 2013 [Online]. Available: https://doi.org/gb8v3k

[14] Y. Garten and R. B. Altman, "Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text," *BMC Bioinformatics*, vol. 10, no. S2, Feb. 2009 [Online]. Available: https://doi.org/df75hq

[15] K. Raja, S. Subramani, and J. Natarajan, "PPInterFinder—a mining tool for extracting causal relations on human proteins from literature," *Database*, vol. 2013, Jan. 2013 [Online]. Available: https://doi.org/gf479b

[16] S. Subramani, R. Kalpana, P. M. Monickaraj, and J. Natarajan, "HPIminer: A text mining system for building and visualizing human protein interaction networks and pathways," *Journal of Biomedical Informatics*, vol. 54, pp. 121–131, Apr. 2015 [Online]. Available: https://doi.org/f7bgnr

[17] M. Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang, "PKDE4J: Entity and relation extraction for public knowledge discovery." *J Biomed Inform*, vol. 57, pp. 320–32, Aug. 2015 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/26277115

[18] H.-M. Müller, K. M. Van Auken, Y. Li, and P. W. Sternberg, "Textpresso Central: a customizable platform for searching, text mining, viewing, and curating biomedical literature," *BMC Bioinformatics*, vol. 19, no. 1, Mar. 2018 [Online]. Available: https://doi.org/gf7rbz

[19] A. Cañada, S. Capella-Gutierrez, O. Rabal, J. Oyarzabal, A. Valencia, and M. Krallinger, "LimTox: a web tool for applied text mining of adverse event and toxicity associations of compounds, drugs and genes," *Nucleic Acids Research*, vol. 45, no. W1, pp. W484–W489, May 2017 [Online]. Available: https://doi.org/gf479h

[20] S. Pletscher-Frankild, A. Pallejà, K. Tsafou, J. X. Binder, and L. J. Jensen, "DISEASES: Text mining and data integration of disease–gene associations," *Methods*, vol. 74, pp. 83–89, Mar. 2015 [Online]. Available: https://doi.org/f3mn6s

[21] Y. Liu, Y. Liang, and D. Wishart, "PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more," *Nucleic Acids Res*, vol. 43, no. W1, pp. W535–W542, Apr. 2015 [Online]. Available: https://doi.org/f7nzn5
[22] J. Zhou and B.-q. Fu, "The research on gene-disease association based on text-mining of PubMed," *BMC Bioinformatics*, vol. 19, no. 1, Feb. 2018 [Online]. Available: https://doi.org/gf479k
[23] J. Kim, H. Kim, Y. Yoon, and S. Park, "LGscore: A method to identify disease-related genes using biological literature and Google data," *Journal of Biomedical Informatics*, vol. 54, pp. 270–282, Apr. 2015 [Online]. Available: https://doi.org/f7bj9c
[24] D. Westergaard, H.-H. Stærfeldt, C. Tønsberg, L. J. Jensen, and S. Brunak, "A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts," *PLoS Comput Biol*, vol. 14, no. 2, p. e1005962, Feb. 2018 [Online]. Available: https://doi.org/gcx747
[25] R. Frijters, M. van Vugt, R. Smeets, R. van Schaik, J. de Vlieg, and W. Alkema, "Literature Mining for the Discovery of Hidden Connections between Drugs, Genes and Diseases," *PLoS Comput Biol*, vol. 6, no. 9, p. e1000943, Sep. 2010 [Online]. Available: https://doi.org/bhrw7x
[26] A. Al-Aamri, K. Taha, Y. Al-Hammadi, M. Maalouf, and D. Homouz, "Analyzing a co-occurrence gene-interaction network to identify disease-gene association," *BMC Bioinformatics*, vol. 20, no. 1, Feb. 2019 [Online]. Available: https://doi.org/gf49nm
[27] J. X. Binder *et al.*, "COMPARTMENTS: unification and visualization of protein subcellular localization evidence," *Database*, vol. 2014, no. 0, pp. bau012–bau012, Feb. 2014 [Online]. Available: https://doi.org/btbm
[28] M. Rastegar-Mojarad, R. K. Elayavilli, D. Li, R. Prasad, and H. Liu, "A new method for prioritizing drug repositioning candidates extracted by literature-based discovery," in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015 [Online]. Available: https://doi.org/gf479j
[29] A. Santos, K. Tsafou, C. Stolte, S. Pletscher-Frankild, S. I. O'Donoghue, and L. J. Jensen, "Comprehensive comparison of large-scale tissue expression datasets," *PeerJ*, vol. 3, p. e1054, Jun. 2015 [Online]. Available: https://doi.org/f3mn6p
[30] A. Junge and L. J. Jensen, "CoCoScore: context-aware co-occurrence scoring for text mining applications using distant supervision," *Bioinformatics*, vol. 36, no. 1, pp. 264–271, Jun. 2019 [Online]. Available: https://doi.org/gf4789
[31] B. Percha and R. B. Altman, "A global network of biomedical relationships derived from text," *Bioinformatics*, vol. 34, no. 15, pp. 2614–2624, Feb. 2018 [Online]. Available: https://doi.org/gc3ndk
[32] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014 [Online]. Available: https://doi.org/gf3xhp
[33] L. J. Jensen, J. Saric, and P. Bork, "Literature mining for the biologist: from information retrieval to biological discovery," *Nat Rev Genet*, vol. 7, no. 2, pp. 119–129, Feb. 2006 [Online]. Available: https://doi.org/bgq7q9
[34] W. W. M. Fleuren and W. Alkema, "Application of text mining in the biomedical domain," *Methods*, vol. 74, pp. 97–106, Mar. 2015 [Online]. Available: https://doi.org/f64p6n
[35] À. Bravo, J. Piñero, N. Queralt-Rosinach, M. Rautschka, and L. I. Furlong, "Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research," *BMC Bioinformatics*, vol. 16, no. 1, Feb. 2015 [Online]. Available: https://doi.org/f7kn8s
[36] E. M. van Mulligen *et al.*, "The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships," *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 879–884, Oct. 2012 [Online]. Available: https://doi.org/f36vn6
[37] R. Bunescu *et al.*, "Comparative experiments on learning information extractors for proteins and their interactions," *Artificial Intelligence in Medicine*, vol. 33, no. 2, pp. 139–155, Feb. 2005 [Online]. Available: https://doi.org/dhztpn
[38] S. Pyysalo *et al.*, "BioInfer: a corpus for information extraction in the biomedical domain," *BMC Bioinformatics*, vol. 8, no. 1, Feb. 2007 [Online]. Available: https://doi.org/b7bhhc
[39] K. Fundel, R. Kuffner, and R. Zimmer, "RelEx–Relation extraction using dependency parse trees," *Bioinformatics*, vol. 23, no. 3, pp. 365–371, Dec. 2006 [Online]. Available: https://doi.org/cz7q4d
[40] J. Li *et al.*, "BioCreative V CDR task corpus: a resource for chemical disease relation extraction," *Database*, vol. 2016, p. baw068, 2016 [Online]. Available: https://doi.org/gf5hfw
[41] M. Krallinger, O. Rabal, S. A. Akhondi, and others, "Overview of the biocreative vi chemical-protein interaction track," in *Proceedings of the sixth biocreative challenge evaluation workshop*, 2017, vol. 1, pp. 141–146 [Online]. Available: https://www.semanticscholar.org/paper/Overview-of-the-BioCreative-VI-chemical-protein-Krallinger-Rabal/eed781f498b563df5a9e8a241c67d63dd1d92ad5
[42] S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski, "Comparative analysis of five protein-protein interaction corpora," *BMC Bioinformatics*, vol. 9, no. S3, Apr. 2008 [Online]. Available: https://doi.org/fh3df7
[43] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998 [Online]. Available: https://doi.org/fwgxrj
[44] D. Xu *et al.*, "DTMiner: identification of potential disease targets through biomedical literature mining," *Bioinformatics*, p. btw503, Aug. 2016 [Online]. Available: https://doi.org/f9nw36
[45] B. Bhasuran and J. Natarajan, "Automatic extraction of gene-disease associations from literature using joint ensemble learning," *PLoS ONE*, vol. 13, no. 7, p. e0200699, Jul. 2018 [Online]. Available: https://doi.org/gdx63f
[46] N. C. Panyam, K. Verspoor, T. Cohn, and K. Ramamohanarao, "Exploiting graph kernels for high performance biomedical relation extraction," *J Biomed Semant*, vol. 9, no. 1, Jan. 2018 [Online]. Available: https://doi.org/gf49nn
[47] N. Warikoo, Y.-C. Chang, and W.-L. Hsu, "LPTK: a linguistic pattern-aware dependency tree kernel approach for the BioCreative VI CHEMPROT task," *Database*, vol. 2018, Jan. 2018 [Online]. Available: https://doi.org/gfhjr6
[48] V. D. Badal, P. J. Kundrotas, and I. A. Vakser, "Text Mining for Protein Docking," *PLoS Comput Biol*, vol. 11, no. 12, p. e1004630, Dec. 2015 [Online]. Available: https://doi.org/gcvj3b
[49] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015 [Online]. Available: https://doi.org/f6v78n
[50] S. Yadav, A. Ekbal, S. Saha, A. Kumar, and P. Bhattacharyya, "Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein–protein interaction," *Knowledge-Based Systems*, vol. 166, pp. 18–29, Feb. 2019 [Online]. Available: https://doi.org/gf4788
[51] Y. Peng, A. Rios, R. Kavuluru, and Z. Lu, "Extracting chemical–protein relations with ensembles of SVM and deep learning models," *Database*, vol. 2018, Jan. 2018 [Online]. Available: https://doi.org/gf479f

[52] S. Liu *et al.*, "Extracting chemical–protein relations using attention-based neural networks," *Database*, vol. 2018, Jan. 2018 [Online]. Available: https://doi.org/gfdz8d
[53] S. Lim and J. Kang, "Chemical–gene relation extraction using recursive neural network," *Database*, vol. 2018, Jan. 2018 [Online]. Available: https://doi.org/gdss6f
[54] Yijia Zhang and Zhiyong Lu, "Exploring Semi-supervised Variational Autoencoders for Biomedical Relation Extraction," arXiv, 1901.06103v1, Jan. 2019 [Online]. Available: https://arxiv.org/abs/1901.06103v1
[55] J. Lee *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, Sep. 2019 [Online]. Available: https://doi.org/ggh5qq
[56] S.-P. Choi, "Extraction of protein–protein interactions (PPIs) from the literature by deep convolutional neural networks with various feature embeddings," *Journal of Information Science*, vol. 44, no. 1, pp. 60–73, Nov. 2016 [Online]. Available: https://doi.org/gcv8bn
[57] Yifan Peng and Zhiyong Lu, "Deep learning for extracting protein-protein interactions from biomedical literature," arXiv, 1706.01556v2, Jun. 2017 [Online]. Available: https://arxiv.org/abs/1706.01556v2
[58] P. Corbett and J. Boyle, "Improving the learning of chemical-protein interactions from literature using transfer learning and specialized word embeddings," *Database*, vol. 2018, Jan. 2018 [Online]. Available: https://doi.org/gf479d
[59] R. Antunes and S. Matos, "Extraction of chemical-protein interactions from the literature using neural networks and narrow instance representation," *Database : the journal of biological databases and curation*. Oxford University Press, Jan-2019 [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6796919/
[60] E. K. Mallory, C. Zhang, C. Ré, and R. B. Altman, "Large-scale extraction of gene interactions from full-text literature using DeepDive," *Bioinformatics*, p. btv476, Sep. 2015 [Online]. Available: https://doi.org/gb5g7b
[61] J. MacArthur *et al.*, "The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog)," *Nucleic Acids Res*, vol. 45, no. D1, pp. D896–D901, Nov. 2016 [Online]. Available: https://doi.org/f9v7cp
[62] D. S. Wishart *et al.*, "DrugBank 5.0: a major update to the DrugBank database for 2018," *Nucleic Acids Research*, vol. 46, no. D1, pp. D1074–D1082, Nov. 2017 [Online]. Available: https://doi.org/gcwtzk
[63] C.-H. Wei, H.-Y. Kao, and Z. Lu, "PubTator: a web-based text mining tool for assisting biocuration," *Nucleic Acids Research*, vol. 41, no. W1, pp. W518–W522, May 2013 [Online]. Available: https://doi.org/f475th
[64] R. Leaman, R. Islamaj Dogan, and Z. Lu, "DNorm: disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, Aug. 2013 [Online]. Available: https://doi.org/f5gj9n
[65] M. Huang, J. Liu, and X. Zhu, "GeneTUKit: a software for document-level gene normalization," *Bioinformatics*, vol. 27, no. 7, pp. 1032–1033, Feb. 2011 [Online]. Available: https://doi.org/dng2cb
[66] C.-H. Wei and H.-Y. Kao, "Cross-species gene normalization by species inference," *BMC Bioinformatics*, vol. 12, no. S8, Oct. 2011 [Online]. Available: https://doi.org/dnmvds
[67] T. C. Wiegers, A. P. Davis, and C. J. Mattingly, "Collaborative biocuration–text-mining development task for document prioritization for curation," *Database*, vol. 2012, no. 0, pp. bas037–bas037, Nov. 2012 [Online]. Available: https://doi.org/gbb3zw
[68] T. Rolland *et al.*, "A Proteome-Scale Map of the Human Interactome Network," *Cell*, vol. 159, no. 5, pp. 1212–1226, Nov. 2014 [Online]. Available: https://doi.org/f3mn6x
[69] S. Razick, G. Magklaras, and I. M. Donaldson, "iRefIndex: A consolidated protein interaction database with provenance," *BMC Bioinformatics*, vol. 9, no. 1, p. 405, 2008 [Online]. Available: https://doi.org/b99bjj
[70] J. Menche *et al.*, "Uncovering disease-disease relationships through the incomplete interactome," *Science*, vol. 347, no. 6224, pp. 1257601–1257601, Feb. 2015 [Online]. Available: https://doi.org/f3mn6z
[71] D. Himmelstein and A. Pankov, "Mining knowledge from MEDLINE articles and their indexed MeSH terms." ThinkLab, 10-May-2015 [Online]. Available: https://doi.org/f3mqwp
[72] D. Himmelstein, L. J. Jensen, M. Smith, K. Fortney, and C. Chung, "Integrating resources with disparate licensing into an open network." ThinkLab, 28-Aug-2015 [Online]. Available: https://doi.org/bfmk
[73] S. Oxenham, "Legal confusion threatens to slow data science," *Nature*, vol. 536, no. 7614, pp. 16–17, Aug. 2016 [Online]. Available: https://doi.org/bndt
[74] S. Carbon, R. Champieux, J. A. McMurry, L. Winfree, L. R. Wyatt, and M. A. Haendel, "An analysis and metric of reusable data licensing practices for biomedical resources," *PLoS ONE*, vol. 14, no. 3, p. e0213090, Mar. 2019 [Online]. Available: https://doi.org/gf5m8v
[75] A. Ratner, B. Hancock, J. Dunnmon, R. Goldman, and C. Ré, "Snorkel MeTaL," in *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning - DEEM'18*, 2018 [Online]. Available: https://doi.org/gf3xk7
[76] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel," *Proc. VLDB Endow.*, vol. 11, no. 3, pp. 269–282, Nov. 2017 [Online]. Available: https://doi.org/ch44
[77] Ye Zhang and Byron Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," arXiv, 1510.03820v4, Oct. 2015 [Online]. Available: https://arxiv.org/abs/1510.03820v4
[78] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," arXiv, 1412.6980v9, Dec. 2014 [Online]. Available: https://arxiv.org/abs/1412.6980v9
[79] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality," arXiv, 1310.4546v1, Oct. 2013 [Online]. Available: https://arxiv.org/abs/1310.4546v1
[80] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching Word Vectors with Subword Information," arXiv, 1607.04606v2, Jul. 2016 [Online]. Available: https://arxiv.org/abs/1607.04606v2
[81] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv, 1301.3781v3, Jan. 2013 [Online]. Available: https://arxiv.org/abs/1301.3781v3
[82] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, "On Calibration of Modern Neural Networks," arXiv, 1706.04599v2, Jun. 2017 [Online]. Available: https://arxiv.org/abs/1706.04599v2
[83] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon, "Accurate Uncertainties for Deep Learning Using Calibrated Regression," arXiv, 1807.00263v1, Jul. 2018 [Online]. Available: https://arxiv.org/abs/1807.00263v1