AGATHA: Automatic Graph-mining And Transformer based Hypothesis generation Approach

Justin Sybrandt * [1], Ilya Tyagin † [1], Michael Shtutman ‡ [2], and Ilya Safro § [1]

[1] School of Computing, Clemson University
[2] Drug Discovery and Biomedical Sciences, University of South Carolina

Abstract

Medical research is risky and expensive. Drug discovery, as an example, requires that researchers efficiently winnow thousands of potential targets to a small candidate set for more thorough evaluation. However, research groups spend significant time and money to perform the experiments necessary to determine this candidate set long before seeing intermediate results. Hypothesis generation systems address this challenge by mining the wealth of publicly available scientific information to predict plausible research directions. We present AGATHA, a deep-learning hypothesis generation system that can introduce data-driven insights earlier in the discovery process. Through a learned ranking criteria, this system quickly prioritizes plausible term-pairs among entity sets, allowing us to recommend new research directions. We massively validate our system with a temporal holdout wherein we predict connections first introduced after 2015 using data published beforehand. We additionally explore biomedical subdomains, and demonstrate AGATHA's predictive capacity across the twenty most popular relationship types. This system achieves best-in-class performance on an established benchmark, and demonstrates high recommendation scores across subdomains. **Reproducibility:** All code, experimental data, and pre-trained models are available online: sybrandt.com/2020/agatha .

1 Introduction

As the rate of global scientific output continues to climb [41], an increasing portion of the biomedical discovery process is becoming a "big data" problem. For instance, the US National Library of Medicine's (NLM) database of biomedical abstracts, *MEDLINE*, has steadily increased the number of papers added per-year, and has added significantly over 800,000 papers every year since 2015 [1]. This wealth of scientific knowledge comes with the overhead cost payed by practitioners who often struggle to keep up with the state-of-the-art.

Buried within the large and growing MEDLINE database are many undiscovered implicit connections — those relationships that are implicitly discoverable, yet have not been identified by the research community. One connection of this type was first proposed and subsequently discovered by Swanson and Smalheiser in the mid-to-late 1980's [37]. Their landmark finding, using only the co-occurrences of keywords across MEDLINE titles, was to establish a connection between fish oil and Raynaud's Syndrome [36]. At that time, it was known that fish oil modified various bodily properties, such as blood viscosity, which were key factors pertaining to Raynaud's syndrome. However, while each explicit relationship was known, the *implicit* relationship was not discovered before Swanson's ARROWSMITH hypothesis generation system identified the connection algorithmically.

Modern advances in machine learning, specifically in the realms of text and graph mining, enable contemporary hypothesis generation systems to identify fruitful new research directions while taking far more

*jsybran@clemson.edu
†ityagin@clemson.edu
‡shtutmanm@sccp.sc.edu
§isafro@clemson.edu

than title co-occurrence rates into account. Modern systems predict missing links on domain specific graphs, such as BioGraph on gene-disease network [26] or MeTeOR on the term-co-occurrence graph [44]. Other systems focus on identifying relevant key terms, similar to Swanson's work, but using modern techniques. For instance, Jha et al. study the evolution of word embedding spaces over time to learn contemporary trends relevant to particular queries [20]. Further work by Jha et al. continues to study the joint evolution of corpora and ontologies within biomedical research [21]. Another approach is to produce visualizations for interpretation by domain scientists [34], such as the closed-source Watson for Drug Discovery [9]. Moliere, our prior hypothesis generation system [39], produces data for scientific interpretation in the form of LDA topic models [7]. Additional work produced heuristically-backed ranking criteria to help automate the analysis process [40].

While prior hypothesis generation systems have been valuable in real-world explorations, such as Swanson's fish-old and Raynaud's syndom finding [36], Watson's discovery of ALS treatments [9], or Moliere's discovery of DDX3 inhibition as a treatment for HIV-associated neurodegenerative disease [5], there remains significant drawbacks to the state of the art. Most systems require significant human oversight to produce useful results [35, 9, 39], or are only tested on very small evaluation sets [20, 21, 15, 30]. Systems still using the "ABC" model of discovery [20, 21, 23], posed by Swanson in 1986 [36], face many known limitations such as reduced scalabiltiy and a bias towards incremental discoveries [32].

To overcome these limitations, we present a new hypothesis generation system that scales to the entirety of biomedical literature, and is backed by efficient deep-learning techniques to enable thousands of queries a minute, enabling new types of queries. This system constructs a new semantic multi-layered graph, and places its millions of nodes into a shared embedding. From there, we use a *transformer encoder* architecture [42] to learn a ranking criteria between regions of our semantic graph and the plausibility of new research connections. Because our graph spans all of MEDLINE, we are able to generate hypotheses from a large range of biomedical subdomains. *Other than our prior work [39], we are unaware of any system that is capable of the same breadth of cross-domain discovery that is also open source, or even just publicly available for comparison.* Because we efficiently pre-process our graph and its embeddings, we can perform hundreds of queries per-second on GPU, which enables new many-to-many recommendation queries that were not previously feasible. Because we replace our heuristically determined ranking criteria from our prior work [40] with a learned ranking criteria, we achieve significantly improved performance, as demonstrated by an increase in benchmark performance using the same training and validation from and ROC AUC of 0.718 [38] to 0.901.

Our contribution:

(1) We introduce a novel approach to construct large semantic graphs that use the granularity of sentences to represent documents. These graphs are constructed using a pipeline of state of the art NLP techniques that have been customized for understanding scientific text, including SciBERT [6] and ScispaCy [28].

(2) We deploy our deep-learning transformer-based model that trained to predict likely connections between term-pairs at scale. This is done by embedding our proposed semantic graph to encode all sentences, entities, n-grams, lemmas, UMLS terms, MeSH terms, chemical identifiers, and SemRep predicates [4] in a common space using the PyTorch-BigGraph embedding [25].

(3) We validate our system using the massive validation techniques presented in [40], and also demonstrate the ability of AGATHA to generalize across biomedical subdomains. For instance, in the scope of "Gene - Cell Function" relationships, our system has a top-10 average precision of 0.83, and a mean-reciprocal-rank of 0.61.

This system is open-source, easily installed, and all prepared data and trained models are available to perform hypothesis queries at sybrandt.com/2020/agatha .

2 Background

Hypothesis Generation Systems. Swanson posited that *undiscovered public knowledge*, those facts that are implicitly available but not explicitly known, would accelerate scientific discovery if an automated system were capable of returning them [37]. His work established what is now known as the "A-B-C" model of literature-based discovery [33]. This formulation follow that a hypothesis generation system, given two terms A and C, should uncover some likely B-terms that explain the quality of a potential $A - B - C$ connection.

This technique fueled Swanson's own system, ARROWSMITH [36], and still forms the backbone of some contemporary successors [23].

Our former approach to address these challenges is posed by the Moliere system [39], and its accompanying plausibility ranking criteria [40]. This system expands on the $A - B - C$ model by describing a range of connection patterns, as represented by an LDA topic model [7], when receiving an A, C query. To do so, the Moliere system first finds a short-path of interactions bridging the $A - C$ connection from within a large semantic graph. This structure includes nodes that correspond to different entity types that are both textual and biomedical, such as abstracts, predicate statements, genes, diseases, proteins, etc. Edges between entities indicate similarity. For instance, an edge may exist between an abstract and all genes discussed within it, or between two proteins that are discussed in similar contexts. Using the short-path discovered within the semantic network between A and C, the Moliere system also reports an LDA topic model [7]. This model summarizes popular areas of conversation pertaining to abstracts identified near to the returned path. As a result, the user can view various fuzzy clusters of entities and the importance of interesting concepts across documents.

To reduce the burden of topic-model analysis on biomedical researchers, the Moliere system is augmented by a range of techniques that automatically quantify the plausibility of the query based on its resulting topic models. Our measures, such as the embedding-based similarity between keywords and topics, as well as network analytic measures based on the topic-nearest-neighbors network, were heuristically backed, and were combined into a meta-measure to best understand potential hypotheses. Using this technique, we both validated the overall performance of the Moliere system, and used it to identify a new gene-treatment target for HIV-associated neurodegenerative disease through the inhibition of DDX3X [3].

Related and Incorporated Technologies.

SemRep [4] is a utility that extracts *predicate statements* in the form of "subject-verb-object" from the entirety of Medline. This utility further classifies its predicate components into the set coded keywords provided by the Unified Medical Language System (UMLS), and a small set of coded verb-types. These UMLS terms provide a way to unify synonyms and acronyms from across medicine. Additionally, all content extracted by SemRep is provided in the Semantic Medical Database (SemMedDB) for direct use.

ScispaCy [28], a version of the popular spaCy text processing library provided by AllenNLP, is designed to properly handle scientific text. Using a deep-learning approach for its part-of-speech tagging, dependency parsing, and entity recognition, this tool achieves state-of-the-art performance on a range of scientific and biomedical linguistic benchmarks. Additionally, this software is optimized sufficiently to operate on each sentence of MEDLINE, which numbers over 188 million as of 2020.

SciBert [6] is a version of the BERT transformer model for scientific language. This model learns representations for each word part in a given sentence. The resulting embeddings for each word part are determined by its relationship to all other word parts. As a result, the output word-part embeddings are highly content-dependent, and homographs, words with the same spelling but different meanings, receive significantly different representations.

FAISS [22], the open-source similarity-search utility, is capable of computing an approximate nearest-neighbors network for huge point clouds. This technique scales to various graph sizes by its modular component set, and we choose PQ-quantization and k-means bucketing to reduce the dimensionality of our sentences, and reduce the search space per-query.

PyTorch-BigGraph (PTBG) [25] is an open-source, large-scale, distributed graph-embedding technique aimed at heterogeneous information networks [31]. These graphs consist of nodes of various types, connected by typed edges. We define each node and relationship type contained in our semantic graph as input to this embedding technique. PTBG distributes edges such that all machines compute on disjoint node-sets. We choose to encode edges through the dot product of transformed embeddings, which we explain in more detail in Section 3.

The Transformer [42] model is built with multi-headed attention. Conceptually, this mechanism works by learning weighted averages per-element of the input sequence, over the entire input sequence. Specifically, this includes three projections of each element's embedding, represented as packed matrices: Q, K, and V. The specific mechanism is defined as follows, with d_k representing the dimensionality of each Q and K embedding:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right)V \tag{1}$$

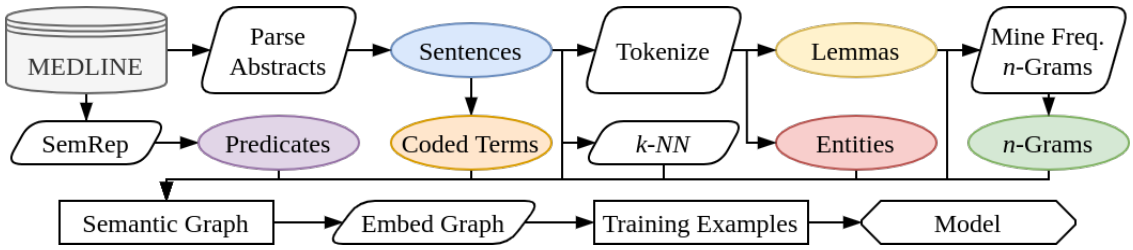

Figure 1: System Diagram of the AGATHA process.

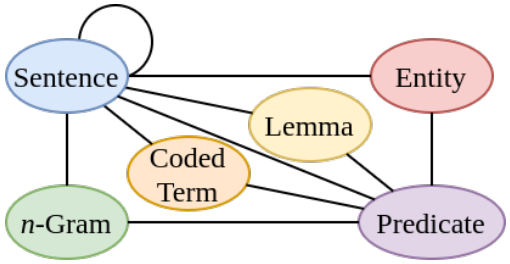

Figure 2: AGATHA multi-layered graph schema.

The "multi-headed" aspect of the transformer indicates that the attention mechanism is applied multiple times per-layer, and recombined to form a joint representation. If $W^{(x)}$ indicates a matrix of learned weights, then this operation is defined as:

$$\begin{aligned} \text{MultiHead}(X) &= [h_1; \ldots; h_k]W^{(4)} \\ \text{where } h_i &= \text{Attention}\left(XW_i^{(1)}, XW_i^{(2)}, XW_i^{(3)}\right) \end{aligned} \tag{2}$$

By using only the encoder half of the transformer model, and by omitting any positional mask or encoding, we apply the self-attention mechanism to understand input sets while reducing the effect of the arbitrary ordering imposed by a sequence model. One encoder layer is defined as:

$$\begin{aligned} \mathcal{E}(X) &= \text{LayerNorm}(FF(\alpha) + \alpha) \\ \text{where } FF(\alpha) &= \max\left(0, \alpha W^{(5)}\right) W^{(6)} \\ \text{and } \alpha &= \text{LayerNorm}(\text{MultiHead}(X) + X) \end{aligned} \tag{3}$$

3 Data Preparation

We propose a significant data processing pipeline (Fig. 1), to convert raw-text sources into a semantic graph (Fig. 2). An embedding of this graph enables our learned ranking criteria.

Text Pre-Processing. We begin with raw MEDLINE XML files [1]. We attempt to extract the paper id (PMID), version, title, abstract text, date of first occurrence, keywords, and publication language. Next, we filter out non-English documents. *In order to validate our system, we additionally discard any document that is dated after January 1st, 2015.*

We split the text of each abstract into sentences. For each sentence, we identify parts-of-speech, dependency tags, and named entities using ScispaCy [28]. The result of this process is a record per-sentence, including the title, that contains all metadata associated with the original abstract, as well as all algorithmically identified annotations.

Using the lemma information of each sentence, we perform n-gram mining in order to identify common phrases that may not have been picked up by entity detection. First, we provide a set of part-of-speech tags we mark as "interesting" from the perspective of n-gram mining. These are: nouns, verbs, adjectives, proper nouns, adverbs, interjections, and "other." We additionally supply a short stopword list, and assert that

[1] At the time of writing, the bulk release at the end of 2019 contained 1,014 files, containing nearly 30-million documents

stop words are uninteresting. Then, for each sentence, we produce the set of n-grams of length two-to-four that both start and end with an interesting lemma. We record any n-gram that achieves an overall support of at least 100. However, we find it necessary to introduce an approximation factor, that an n-gram must have a minimum support of five within a datafile for those occurrences to count.

Semantic Graph Construction. After splitting sentences, while simultaneously identifying lemmas, entities and n-grams, we can begin constructing the semantic graph. We begin this process by creating edges between similar sentences. The simplest edge we add is that between two adjacent sentences from the same abstract. For instance, sentence i in abstract A will produce edges to A_{i-1} and A_{i+1}, with the paper title serving as A_0.

To capture edges between similar sentences in different abstracts, we compute an approximate-nearest-neighbors network on the set of sentence embeddings. We derive these embeddings from the average of the final hidden layer of the SciBert [2] NLP model for scientific text [6]. This 768-dimensional embedding captures context-sensitive content regarding each word in each sentence.

However, we have over 155-million sentences in the 2015 validation instance of AGATHA, which makes performing a nearest-neighbors search per-sentence (typically $\mathcal{O}(n^2d)$) computationally difficult. Therefore, we leverage FAISS to perform dimensionality reduction, as well as approximate-nearest neighbors, in a distributed setting. First, we collect a one-percent sample of all embeddings on a single machine, wherein we perform product quantization (PQ) [18]. This technique learns an efficient bit representation of each embedding. We use 96-quantizers, and each considers a disjoint an 8-dimensional chunk of the 768-dimensional SciBert embeddings. Each quantizer then learns to map its input real-valued chunk into output 8-bit codes, such that similar input chunks receive output codes with low hamming distance.

Still using the 1% sample on one machine, FAISS performs k-Means over PQ codes in order to partition the reduced space into self-similar buckets. By storing the centroid of each bucket, we can later select a relevant sub-space pertaining to each input query, dramatically reducing the search space. We select 2048 partitions to divide the space, and when performing a query, each input embedding is compared to all embeddings residing in the 16 most-similar buckets.

Once the PQ quantizers and k-means buckets are determined, the initial parameters are distributed to each machine in the cluster. Every sentence can be added to the FAISS nearest-neighbors index structure in parallel, and then the reduced codes and buckets can be merged in-memory on one machine. We again distributed the nearest-neighbors index, now containing all 155-million sentence codes, to each machine in the cluster. In parallel, these machines can identify relevant buckets per-point, and record their 25 approximate nearest-neighbors. If we have m machines, each with p cores, and search $q = 16$ of the $b = 2048$ buckets-per-query, we reduce complexity for identifying all nearest-neighbors from $\mathcal{O}(dn^2)$ to $\mathcal{O}\left({}^{qdn^2}/_{32bpm}\right)$.

We additionally add simpler sentence-occurrence edges for lemmas, n-grams and entities. In each case, we produce an edge between s and x provided that lemma, entity, n-gram, or metadata-keyword x occurs in sentence s. The last node type is SemRep predicates [4]. Each has associated metadata, such as the sentence in which it occurred, its raw text, and its relevant UMLS coded terms. For each unique subject-verb-object triple, we create a node in the semantic graph. We then create edges from that node to each relevant sentence, keyword, lemma, entity, and n-gram. Our overall graph consists of *184-million nodes and 12.3-billion edges.*

Graph Embedding. We utilize the PyTorch-BigGraph (PTBG) embedding utility to perform a distributed embedding of the entire network [25]. PTBG learns typed embeddings, and we define node types corresponding to each presented in our semantic graph schema. Each undirected edge in our graph schema is also coded as two directional edges of types $x \to y$ and $y \to x$.

We explore two different embedding dimensionalities: 256 and 512. When computing both embeddings, we specify for edges to be encoded via the dot-product of nodes, and for relationship types to be encoded using a learned translation per-type. We generate a total of 100 negative samples per edge, 50 chosen from nodes within each batch, and 50 chosen from nodes within the corresponding partitions. Dot products between embeddings are learned using the supplied softmax loss, with the first dimension of every embedding acting as a bias unit.

Formally, if an edge ij exists between nodes i and j of types t_i and t_j respectively, then we learn an embedding function $e(\cdot)$ that is used to create a score for ij by projecting each node into $\mathbb{R}^N$ where N

[2] We specifically use the pre-trained "scibert-scivocab-uncased" model, which was trained on over 1.14-million full-text papers.

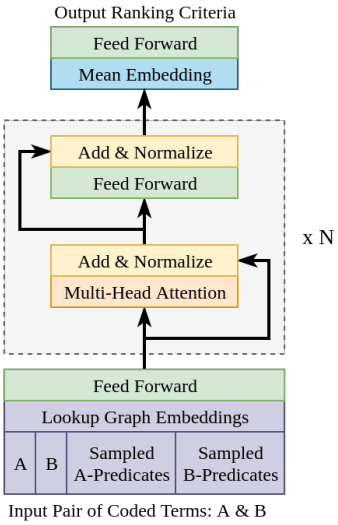

Figure 3: AGATHA ranking transformer encoder.

is a predetermined embedding dimensionality. In our experiments we consider $N = 256$ and 512. This embedding function uses the typed translation vector $T^{(t_i t_j)} \in \mathbb{R}^N$ that is shared for all edges of the same type as ij. This score is defined as:

$$s(ij) = e(i)_1 + e(j)_1 + T_1^{(t_i t_j)} + \sum_{k=2}^{N} e(i)_k \left(e(j)_k + T_k^{(t_i t_j)} \right) \tag{4}$$

Then, for each edge ij, we generate 100 negative samples in the form $x_n^{(ij)} y_n^{(ij)}$. Their scores are compared to that of the positive sample using the following loss function, which indicates the component of overall loss corresponding to edge ij:

$$\text{GraphLoss}_{ij} = -s(ij) + \log \sum_{n=0}^{100} \exp \left(s \left(x_n^{(ij)} y_n^{(ij)} \right) \right) \tag{5}$$

Training Data. In order to learn what makes a plausible biomedical connection, we collect the set of published connections present in our pre-2015 training set. For this, we turn to the Semantic Medical Database (SemMedDB), which contains over 19-million pre-2015 SemRep [4] predicates parsed from all of MEDLINE. A SemRep predicate is a published subject-verb-object triple that is identified algorithmically. In lieu of a true data set of attempted hypotheses, we can train our model on these published connections. However, this approach comes with some drawbacks. Firstly, SemRep predicates are defined on the set of UMLS terms, which will restrict our system to only those entities that have been coded. This limitation is acceptable given size size of UMLS, and presence of existing benchmarks defined among UMLS terms [40]. Secondly, the predicate set is noisy, and may contain entries that are incorrect or obsolete, as well as algorithmically introduced inaccuracies. However, we find at scale that these sources of noise do not overwhelm the useful signal present within SemMedDB.

4 Ranking Plausible Connections

We train a model to rank published SemRep [4] predicates above noisy negative samples using the transformer architecture [42]. To do so we first formulate a predicate with subject α and object β for input into the model. Those predicates that are collected from SemRep are "positive samples" (PS). The function $\Gamma(\cdot)$ indicates the set of neighbor predicates that include a term as either a subject or object. We represent the $\alpha\beta$ predicate as a set with elements that include both terms, as well as a fixed-size sample with-replacement

of size $s = 15$ of each node's non-shared predicates:

$$\begin{aligned} \text{PS}_{\alpha\beta} &= \left\{\alpha, \beta, \gamma_1^{(\alpha)}, \ldots, \gamma_s^{(\alpha)}, \gamma_1^{(\beta)}, \ldots, \gamma_s^{(\beta)}\right\} \\ \text{where } \gamma_i^{(\alpha)} &\sim \{\Gamma(\alpha) - \Gamma(\beta)\}, \text{ and } \gamma_i^{(\beta)} \sim \{\Gamma(\beta) - \Gamma(\alpha)\} \end{aligned} \tag{6}$$

Negative Samples We cannot learn to rank positive training examples in isolation. Instead, we first generate negative samples to accompany each published predicate. This include two types of samples: scrambles and swaps. Both are necessary, as we find during training that the easier-to-distinguish scrambles aid early convergence, while the swaps require the model to understand the biomedical concepts encoded by the semantic graph embedding.

The negative scramble (NScr) selects two arbitrary terms x and y, as well as $2s$ arbitrary predicates from the set of training data. While we enforce that x and y do not share a predicate, we do not enforce any relationship between the sampled predicates and these terms. Therefore these samples are easy to distinguish from positive examples. If T denote all positive-set terms, and P denotes all predicates, then a negative scramble associated with positive sample $\alpha\beta$ is notated as:

$$\begin{aligned} \text{NScr}_{\alpha\beta} &= \{x, y, \gamma_1, \ldots, \gamma_{2s}\} \\ \text{where } x, y &\sim T, \text{ and } \gamma_i \sim P \\ \text{s.t. } \Gamma(x) &\cap \Gamma(y) = \emptyset \end{aligned} \tag{7}$$

The negative swap (NSwp) selects two arbitrary terms, but samples the associated predicates in the same manner as the positive sample. Therefore, the observed term-predicate relationship will be the same for each half of this negative sample (α and $\gamma_i^{(\alpha)}$). This sample requires the model to learn that some $\alpha\beta$ pairs should not go together, and this will require an understanding of the relationships between biomedical terms. A negative scramble associated with $\alpha\beta$ is notated as:

$$\begin{aligned} \text{NSwp}_{\alpha\beta} &= \left\{x, y, \gamma_1^{(x)}, \ldots, \gamma_s^{(x)}, \gamma_1^{(y)}, \ldots, \gamma_s^{(y)}\right\} \\ \text{where } \gamma_i^{(x)} &\sim \{\Gamma(x) - \Gamma(y)\}, , \text{ and } \gamma_i^{(y)} \sim \{\Gamma(y) - \Gamma(x)\} \\ \text{s.t. } \Gamma(x) &\cap \Gamma(y) = \emptyset \end{aligned} \tag{8}$$

Objective. We minimize the margin ranking loss between each positive sample and all associated negative samples. The contribution of positive sample $\alpha\beta$ to the overall loss is defined as:

$$\begin{aligned} \mathcal{L}(\alpha, \beta) = \sum_{i=0}^{n} L\left(\text{PS}_{\alpha\beta}, \text{Nscr}_{\alpha\beta}^{(i)}\right) + \sum_{j=0}^{n'} L\left(\text{PS}_{\alpha\beta}, \text{Nswp}_{\alpha\beta}^{(j)}\right) \\ \text{where } L(p, n) = \max\left(0, m - \mathcal{H}(p) + \mathcal{H}(n)\right) \end{aligned} \tag{9}$$

Here $n = 10$ denotes the number of negative scrambles, $n' = 30$ is the number of negative swaps, $m = 0.1$ is the desired margin between positive and negative samples, and $\mathcal{H}$ is the learned function that produces a ranking criteria given two terms and a sample of predicates.

Model. Using the transformer encoder summarized in Section 2, as well as the semantic graph embedding, we construct our model. If $e(x)$ represents the semantic graph embedding of x, FF represents a feed-forward layer, and $\mathcal{E}$ represents an encoder layer, then our model $\mathcal{H}$ is defined as:

$$\begin{aligned} \mathcal{H}(X) &= \text{sigmoid}(\mathcal{M}W) \\ \mathcal{M} &= \frac{1}{|X|} \sum_{x_i \in X} E_N(FF(e(x_i))) \\ E_{i+1}(x) &= \mathcal{E}(E_i(x)), \text{ and } E_0(x) = x \end{aligned} \tag{10}$$

Here $N = 4$ represents the number of encoder layers, and W indicates the learned weights associated with the final ranking projection. By averaging the transformer output over the input sequence X, then projecting that result down to a single real value with W, and applying the sigmoid function, we produce an output per-predicate in the unit interval. This function is depicted in Figure 4. The supplemental information containing training parameters and additional model detail.

5 Validation

Testing hypothesis generation, in contrast to information retrieval, is difficult as ultimately these systems are intended to discover information that is unknown to even those designing them [45]. A thorough evaluation would require a costly process wherein scientists explore automatically posed hypotheses. Instead, we perform a historical validation, in a manner similar to that performed in [40, 38]. This method enables large-scale evaluation of many biomedical subdomains almost instantly, but cannot truly tell us how our system will perform in a laboratory environment.

Comparison with Heuristic-Based Ranking. We begin by comparing the performance numbers obtained through our proposed learned ranking criteria with other ranking methods posed in [40]. Specifically, the Moliere system presents experimental numbers for various training-data scenarios for the same 2015 temporal holdout as used in this work [38]. For a direct comparison, we use our proposed method to rank the same set of positive and negative validation examples.

Comparison by Subdomain Recommendation. As mentioned in [16], the Moliere validation set has limitations. We improve this set by expanding both the quantity and diversity of considered term pairs, as well as evaluating AGATHA through the use of all-pairs recommendation queries within popular biomedical subdomains. As a result, this comparison effectively uses subdomain-specific negative examples, which makes for a harder benchmark than that presented in the Moliere work. It is worth nothing that these all-pairs searches are made possible by the very efficient neural-network inference within AGATHA, and would not be as computationally efficient in the Moliere shortest-path and topic-modeling approach.

This analysis begins by extracting *semantic types* [2], which categorize each UMLS term per-predicate into one of 134 categories, including "Lipid," "Plant," or "Enzyme." From there, we can group $\alpha\beta$ predicate-term pairs by types t_α and t_β. We select the twenty predicate type pairs with the most popularity in the post-2015 dataset, and within each type we identify the top-100 predicates with the most rapid non-decreasing growth of popularity determined by the number of abstracts containing each term-pair per year. These predicates form the positive class of the validation set. We form the rest of the subdomain's validation set by recording all possible undiscovered pairs of type $t_\alpha t_\beta$ from among the UMLS terms in the top-100 predicates. We then rank the resulting set by the learned ranking criteria, and evaluate these results using a range of metrics.

Metrics. The first metrics we consider are typical for determining a classification threshold: the area under the receiver-operating-characteristic curve (AUC ROC) and the area under the precision-recall curve (AUC PR). We additionally provide recommendation system metrics, such as top-k precision (P.@k), average precision (AP.@k), and overall reciprocal rank (RR). Top-k precision is simply the number of published term-pairs appearing in the first k elements of the ranked list, divided by k. Top-k *average* precision weights each published result by its location in the front of the ranked list. The reciprocal rank is the inverse of the rank of the first published term pair.

The above recommender system metrics all consider the single many-to-many query within a biomedical subdomain. However, this same result can be interpreted as a set of one-to-many recommendation queries. Doing so enables us to compute the mean average precision(MAP.@k), and mean-reciprocal rank(MRR.@k) for the set of recommendations. A high MRR within a domain indicates that the researcher should expect to see a useful result within the first few results. A high MAP indicates that out of the top k results, more of them are useful. These metrics, taken together, should influence biomedical researchers when exploring the results of a one-to-many query.

6 Results

We compare the performance of AGATHA against Moliere, as presented in [38]. In that work, multiple trained instances of Moliere rank a benchmark set of positive and negative potential connections using a range of criteria defined in [40]. These Moliere instances each use different datasets published prior to 2015 in order to perform hypothesis queries, of which we focus on two: all of MEDLINE (Moliere: MEDLINE), and all of PubMedCentral (Moliere: Full Text). The former instance represents a system trained on the same raw data as the AGATHA system presented here, while the latter represents a system trained on all publicly available full-text papers provided by the NLM released in the same date range.

The prior work establishes that the Moliere topic-modeling approach is improved by the additional information made available by full-text papers, but at a overwhelming 45x runtime penalty. These quality

Type	Training %	Training Rank	AUC PR	AUC ROC	RR	P.@ 10	P.@ 100	AP.@ 10	AP.@ 100	MAP.@ 10	MAP.@ 100	MRR.@ 10	MRR.@ 100
gngm, celf	0.29	74	0.44	0.62	1.00	0.50	0.47	0.83	0.54	0.57	0.56	0.61	0.61
gngm, neop	0.35	61	0.34	0.65	0.50	0.50	0.43	0.54	0.47	0.46	0.41	0.52	0.52
aapp, neop	0.35	62	0.20	0.62	0.33	0.30	0.26	0.34	0.28	0.40	0.35	0.46	0.47
gngm, cell	0.43	42	0.19	0.72	0.25	0.30	0.17	0.27	0.21	0.35	0.32	0.38	0.38
aapp, cell	0.67	26	0.19	0.63	0.50	0.20	0.17	0.36	0.21	0.34	0.33	0.37	0.38
aapp, gngm	1.05	13	0.17	0.68	1.00	0.50	0.22	0.61	0.31	0.36	0.27	0.39	0.40
cell, aapp	1.59	4	0.17	0.67	0.14	0.10	0.19	0.14	0.18	0.35	0.32	0.40	0.41
gngm, gngm	0.50	37	0.17	0.66	1.00	0.40	0.20	0.77	0.37	0.31	0.26	0.33	0.34
orch, gngm	0.41	49	0.16	0.69	0.05	0.00	0.22	0.00	0.21	0.33	0.27	0.34	0.36
aapp, dsyn	0.67	25	0.15	0.69	0.33	0.20	0.24	0.28	0.22	0.34	0.27	0.37	0.38
gngm, dsyn	0.21	97	0.15	0.71	0.50	0.40	0.24	0.59	0.32	0.29	0.23	0.30	0.31
bpoc, aapp	1.06	12	0.14	0.67	1.00	0.20	0.18	0.70	0.28	0.35	0.30	0.38	0.39
bacs, gngm	0.29	73	0.12	0.67	0.33	0.10	0.14	0.33	0.19	0.26	0.24	0.29	0.30
bacs, aapp	0.73	22	0.12	0.68	0.17	0.30	0.14	0.28	0.18	0.27	0.24	0.30	0.32
dsyn, humn	7.02	1	0.11	0.64	0.05	0.00	0.10	0.00	0.10	0.27	0.25	0.29	0.31
aapp, aapp	1.57	5	0.11	0.69	1.00	0.20	0.11	0.67	0.25	0.28	0.24	0.32	0.33
gngm, aapp	0.40	52	0.11	0.71	1.00	0.20	0.11	0.61	0.22	0.23	0.21	0.25	0.26
phsu, dsyn	0.76	20	0.10	0.61	0.04	0.00	0.14	0.00	0.11	0.27	0.20	0.30	0.31
dsyn, dsyn	1.35	6	0.09	0.62	0.17	0.20	0.12	0.19	0.15	0.22	0.18	0.25	0.27
topp, dsyn	1.19	9	0.09	0.64	0.10	0.10	0.17	0.10	0.12	0.28	0.22	0.30	0.31

Table 1: AGATHA-512. Above are hypothesis prediction results on biomedical sub-domains. Indicated along with performance numbers are the percentage of training data (pre-2015 predates) as well as the training-data popularity rank out of 6396, with 1 being most popular. Metrics described in detail in Section 5.

System Instance	ROC AUC	PR AUC
Moliere: Medline	0.718	0.820
Moliere: Full Text	0.795	0.778
AGATHA-256	0.826	0.895
AGATHA-512	**0.901**	**0.936**

Table 2: Benchmark comparison between Moliere and AGATHA on the same benchmark.

results are reproduced in Table 2, and we include additional results for the AGATHA system when evaluated on only abstracts, and exactly the same set of predicates. We observe that the AGATHA system, when trained with 512-dimensional graph embeddings, improves upon Moliere: Medline by 25% and Moliere: Full Text by 13%. Importantly, this increase in quality comes at an overwhelming *decrease* in runtime, with the wall time per-query dropping from minutes to milliseconds, due to the introduction of the deep-learning approach.

To extend the validation beyond the above results, provided that we can now generate thousands of hypothesis per-minute, we explore the capacity of our deep-learning ranking criteria to perform hypothesis recommendation within various many-to-many queries across different biomedical sub-domains. These results, displayed in Table 1, list the 20 predicate types with the most popularity following 2015. Due to space limitations, we present predicate types using NLM semantic type codes [2]. All numbers are reported from the AGATHA-512 model.

We observe that the (Gene)→(Cell Function)(gngm, celf) predicate type, is the easiest predicate type for AGATHA-512 to recommend, even though connections of this type only account for 0.29% of the training data. Of the top-10 recommendations the highest ranked is a valid connection and half are valuable. When performing a one-to-many query within this type of connection, we observe 85% of all top-10 suggestions to be useful on average, and that a useful result occurs typically within the first two recommendations. We see similar performance in the (Gene)→(Neoplastic Process) (gngm, neop) and (Amino Acid, Peptide, or Protein)→(Neoplastic Process) (aapp, neop) sub-domains. Interestingly, there appears to be little correlation between the popularity of a predicate type in the training data and the quality of the resulting recommendations. This result enforces the idea of AGATHA as a *general-purpose* biomedical hypothesis

generation system.

Of the 20-most-popular predicate subdomains considered, AGATHA-512 has the most difficulty with the (Therapeutic or Preventive Procedure)→(Disease or Syndrome)(topp, dsyn). In this subdomain, the the highest ranked positive predicate is ranked tenth, and only twelve of the top-100 suggestions are useful. Still, in a one-to-many query, we expect about one-in-ten recommended predicates to be useful, and for the top-3 predicates to contain a useful result. While the lower-performing subdomains are significantly harder for AGATHA-512 than the top few, we note that even a low-precision tool can be useful for aiding the biomedical discovery process. Furthermore, these difficult subdomains are still ranked significantly better than random chance, and even better than many of the classical ranking measures presented in [40]. Using this information, future work may wish to fine-tune the AGATHA method to a specific subdomain for improved performance.

7 Lessons Learned and Open Problems

Result Interpretability. While deep-learning models are notoriously hard for human decision makers to interpret, we find that biomedical researchers still need to understand how a result was produced in order to act on model predictions. However, we cannot leave the entire analysis up for human judgement, as this drastically reduces the benefits of "automatic" hypothesis generation. To walk the narrow edge between these conflicting objectives, we implement both an automatic ranking component, as well as a more interpretable topic-model query system. We find that these tools serve different functions during different times of the discovery process.

At first, a researcher may be considering a wide range of potential research directions, such as during the candidate selection phase of the drug-discovery process. This often requires assembling hundreds (or thousands) of target ingredients, compounds, genes, or deceases, and determining whether elements of this large set have a relationship to an item of interest. For instance, when we evaluated HIV-associated Neurodegenerative Disease, we explored over 40,000 potential human genes [3]. This component of the discovery process fits nicely into the deep-learning ranking and recommendation system proposed here, especially when the target set is so large that a manual literature review may prove costly.

Once a candidate set of targets has been winnowed from the large target set, the researcher will prioritize interpretability. However, the candidate set is typically orders of magnitude smaller than the target set. Therefore, we can afford to run more costly-yet-interpretable routines, even if these routines do not provide any form of "automatic" analysis. At this stage, we switch from our deep-learning ranking method to the topic-modeling approach similar to that presented in Moliere [39]. This process finds a path within our semantic network containing the textual information necessary to describe a potential connection. We present that path along with the set of relevant sentences, as well as a visualization of the topic model built from those sentences. Researchers can explore the sets of entities that are frequently mentioned together in order to expand their mental models of each hypothesis's quality.

Datasets and Expandability. When discussing hypothesis generation systems with prospective adopters in the biomedical community, we often are asked to include specific datasets that has domain-relevance to an individual's research direction. For instance, the set of clinical trials, internal experimental findings, or a database of chemicals.

We designed AGATHA to be easily extendable. Domain scientists can easily supply new graph datasets as a collection of TSV files and by making minor changes to the PTBG configuration. Furthermore, new textual sources can be merged into the pipeline with straightforward modification to the python data-processing pipeline. In contrast to the graph and text sources, it is not currently clear how to incorporate experimental data into the AGATHA system. This challenge arises from the many forms experimental data can take. In the case where an experiment can be reformulated as a network, such as converting the gene-expression matrix into a gene-to-gene network, these results can trivially be introduced as new edges. Other experimental results, such as many clinical trials, include a thorough summary of that trial's findings. These may be introduced as a combination of textual and graph-based sources, including both the description text, as well as any links to known publications that reference the trial. Importantly, we do not find a "one size fits all" solution for experimental data, and more work should explore the costs and benefit associated with various datasets.

8 Related Work

Foster et al. [13] identify a series of common successful research strategies often used by scientists. In doing so they demonstrate that high-risk and innovative strategies are uncommon among the scientific community in general. It follows that the field of hypotheses generation obeys similar rules. Many systems have found success using algorithmic techniques that approximate these common research strategies by studying term co-occurrences [19, 17, 43], or predicting links with a graph of biomedical entities [29, 11]. While the Foster's model of research strategies has proven to be useful, the mechanisms involved in complex scientific discoveries remain unexplored.

Unsurprisingly, we find that hypothesis generation systems utilize algorithmic techniques in a range of complexity that is analogous to these human research strategies. The first hypothesis generation system, ARROWSMITH, presents the ABC model of automatic discovery [36]. This technique identifies a list of terms that are anticipated to help explain a connection between two terms of interest. This basic algorithm remains in some modern systems, such as [23]. However, ABC-based techniques have significant limitations [32], including their similarity metrics defined on heuristically determined term lists, as well as their reliance on manual validation processes. As a result, ABC systems are know to be biased towards finding incremental discoveries [24].

A completely different strategy of performing LBD is proposed by Spangler et al. in [35]. To explore the p53 kinase, the authors use neighborhood graphs constructed from entity co-occurrence rates. The approach relies on domain experts and requires manual oversight to provide MEDLINE search queries, and to prune redundant terms, but produces promising results. In [10] the authors demonstrate that this technique can identify kinase NEK2 as an inhibitor of p53, and in [5] a similar scientist-in-the-loop technique identifies a number of RNA-binding proteins associated with ALS.

A significant step beyond ABC and human-assisted techniques is to incorporate a domain specific datasets. Bipartite graphs, such as the gene-disease [27] or the term-document [14] networks, are frequent choices. These systems usually aim to perform a number of graph traversals between node-pairs in order to rank the most viable options. However, the number of generated paths may be prohibitively large, which reduces ranking quality [15] To address this problem, Gopalakrishnan proposes two-stage filtering through a "single-class classifier" which is able to prune up to 90% hypotheses prior to the ranking scheme [14]

One recent approach is to use deep learning models to help extract viable biomedical hypotheses. Sang et al. [30] describe GrEDeL, a way to generate new hypotheses using knowledge graphs obtained from predicate triples in the form of subject, verb, object. This approach finds all possible paths between a given drug and decease, provided those paths include a particular target entity. Then these paths are evaluated using a LSTM model that captures features related to drug-disease associations. While the GrEDeL system is successful at identifying some novel drug-disease relationships, this approach has some important trade-offs: (1) Their proposed model is trained using SemRep graph traversals as a sequence, which the authors note is a highly noisy dataset. Furthermore, multiple redundant and similar paths exist within their dataset, which decrease the quality of their validation holdout set. The AGATHA system overcomes this limitation by leveraging node neighborhoods in place of paths. (2) Their knowledge graph is constructed exclusively from predicates mined from MEDLINE abstracts using SemRep. This process affects the model quality significantly and, being the only resource of knowledge, it requires careful manual filtering of false positive and isolated predicates. (3) The GrEDeL LSTM model is trained to only discover drug-disease associations, and does not generalize to other biomedical subdomains. (4) This approach embeds their predicate knowledge graph using the TransE method [8], which supposes that relationships can be modeled as direct linear transformations. When using the large number of relationship types present in SemRep, this assumption greatly reduces the useful variance in the resulting node embeddings.

9 Conclusions

This work presents AGATHA, a deep-learning biomedical hypothesis generation system, which can accelerate discovery by learning to detect useful new research ideas from existing literature. This technique enables domain scientists to keep pace with the accelerating rate of publications, and to efficiently extract implicit connections from the breadth of biomedical research. By constructing a large semantic network, embedding that network, and then training a transformer-encoder deep-learning model, we can learn a ranking criteria

that prioritizes plausible connections. We validate this ranking technique by constructing an instance of the AGATHA system using only data published prior to January 1st 2015. This system then evaluates both a benchmark of predicates established from prior work [38], and performs recommendation in twenty popular biomedical subdomains. The result is state-of-the art prediction quality on the 2015 benchmark, as well as strong performance across a range of subdomains. The AGATHA system is open-source and written entirely in Python and PyTorch, which enable to be easily used or adapted anywhere. We release both the 2015 validation system, as well as an up-to-date 2019 system to accelerate the broader community of biomedical sciences.

References

[1] Citations added to medline by fiscal year.

[2] Semantic types.

[3] Marina Aksenova, Justin Sybrandt, Biyun Cui, Vitali Sikirzhytski, Hao Ji, Diana Odhiambo, Matthew D Lucius, Jill R Turner, Eugenia Broude, Edsel Peña, et al. Inhibition of the dead box rna helicase 3 prevents hiv-1 tat and cocaine-induced neurotoxicity by targeting microglia activation. *Journal of Neuroimmune Pharmacology*, pages 1–15, 2019.

[4] Patrick Arnold and Erhard Rahm. Semrep: A repository for semantic mapping. *Datenbanksysteme für Business, Technologie und Web (BTW 2015)*, 2015.

[5] Nadine Bakkar, Tina Kovalik, Ileana Lorenzini, Scott Spangler, Alix Lacoste, Kyle Sponaugle, Philip Ferrante, Elenee Argentinis, Rita Sattler, and Robert Bowser. Artificial intelligence in neurodegenerative disease research: use of ibm watson to identify additional rna-binding proteins altered in amyotrophic lateral sclerosis. *Acta neuropathologica*, 135(2):227–247, 2018.

[6] Iz Beltagy, Arman Cohan, and Kyle Lo. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[7] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[8] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[9] Ying Chen, JD Elenee Argentinis, and Griff Weber. Ibm watson: how cognitive computing can be applied to big data challenges in life sciences research. *Clinical therapeutics*, 38(4):688–701, 2016.

[10] Byung-Kwon Choi, Tajhal Dayaram, Neha Parikh, Angela D Wilkins, Meena Nagarajan, Ilya B Novikov, Benjamin J Bachman, Sung Yun Jung, Peter J Haas, Jacques L Labrie, et al. Literature-based automated discovery of tumor suppressor p53 phosphorylation and inhibition by nek2. *Proceedings of the National Academy of Sciences*, 115(42):10666–10671, 2018.

[11] Lauri Eronen and Hannu Toivonen. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC bioinformatics*, 13(1):119, 2012.

[12] W.A. et al. Falcon. Pytorch lightning. `https://github.com/PytorchLightning/pytorch-lightning`, 2019.

[13] Jacob G Foster, Andrey Rzhetsky, and James A Evans. Tradition and innovation in scientists research strategies. *American Sociological Review*, 80(5):875–908, 2015.

[14] Vishrawas Gopalakrishnan, Kishlay Jha, Guangxu Xun, Hung Q Ngo, and Aidong Zhang. Towards self-learning based hypotheses generation in biomedical text domain. *Bioinformatics*, 34(12):2103–2115, 2018.

[15] Vishrawas Gopalakrishnan, Kishlay Jha, Aidong Zhang, and Wei Jin. Generating hypothesis: Using global and local features in graph to discover new knowledge from medical literature. In *Proceedings of the 8th International Conference on Bioinformatics and Computational Biology, BICOB*, pages 23–30, 2016.

[16] Sam Henry. Indirect relatedness, evaluation, and visualization for literature based discovery. 2019.

[17] Dimitar Hristovski, Carol Friedman, Thomas C Rindflesch, and Borut Peterlin. Exploiting semantic relations for literature-based discovery. In *AMIA annual symposium proceedings*, volume 2006, page 349. American Medical Informatics Association, 2006.

[18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[19] Rob Jelier, Martijn J Schuemie, Antoine Veldhoven, Lambert CJ Dorssers, Guido Jenster, and Jan A Kors. Anni 2.0: a multipurpose text-mining tool for the life sciences. *Genome biology*, 9(6):R96, 2008.

[20] Kishlay Jha, Guangxu Xun, Yaqing Wang, Vishrawas Gopalakrishnan, and Aidong Zhang. Conceptsbridges: Uncovering conceptual bridges based on biomedical concept evolution. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1599–1607, 2018.

[21] Kishlay Jha, Guangxu Xun, Yaqing Wang, and Aidong Zhang. Hypothesis generation from text based on co-evolution of biomedical concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 843–851, 2019.

[22] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[23] Yong Hwan Kim and Min Song. A context-based abc model for literature-based discovery. *PloS one*, 14(4), 2019.

[24] Ronald N Kostoff, Joel A Block, Jeffrey L Solka, Michael B Briggs, Robert L Rushenberg, Jesse A Stump, Dustin Johnson, Terence J Lyons, and Jeffrey R Wyatt. Literature-related discovery. *Annual review of information science and technology*, 43(1):1–71, 2009.

[25] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA, 2019.

[26] Anthony ML Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. Biograph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome biology*, 12(6):R57, 2011.

[27] Chun-Chi Liu, Yu-Ting Tseng, Wenyuan Li, Chia-Yu Wu, Ilya Mayzus, Andrey Rzhetsky, Fengzhu Sun, Michael Waterman, Jeremy JW Chen, Preet M Chaudhary, et al. Diseaseconnect: a comprehensive web server for mechanism-based disease–disease connections. *Nucleic acids research*, 42(W1):W137–W146, 2014.

[28] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*, 2019.

[29] Murali K Pusala, Ryan G Benton, Vijay V Raghavan, and Raju N Gottumukkala. Supervised approach to rank predicted links using interestingness measures. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1085–1092. IEEE, 2017.

[30] Shengtian Sang, Zhihao Yang, Xiaoxia Liu, Lei Wang, Hongfei Lin, Jian Wang, and Michel Dumontier. Gredel: A knowledge graph embedding based method for drug discovery from biomedical literatures. *IEEE Access*, 7:8404–8415, 2018.

[31] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2016.

[32] Neil R Smalheiser. Literature-based discovery: Beyond the abcs. *Journal of the American Society for Information Science and Technology*, 63(2):218–224, 2012.

[33] Neil R Smalheiser. Rediscovering don swanson: The past, present and future of literature-based discovery. *Journal of Data and Information Science*, 2(4):43–64, 2017.

[34] Scott Spangler. *Accelerating Discovery: Mining Unstructured Information for Hypothesis Generation*. Chapman and Hall/CRC, 2015.

[35] Scott Spangler, Angela D Wilkins, Benjamin J Bachman, Meena Nagarajan, Tajhal Dayaram, Peter Haas, Sam Regenbogen, Curtis R Pickering, Austin Comer, Jeffrey N Myers, et al. Automated hypothesis generation based on mining scientific literature. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1877–1886, 2014.

[36] Don R Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.

[37] Don R Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118, 1986.

[38] Justin Sybrandt, Angelo Carrabba, Alexander Herzog, and Ilya Safro. Are abstracts enough for hypothesis generation? In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1504–1513, 2018.

[39] Justin Sybrandt, Michael Shtutman, and Ilya Safro. Moliere: Automatic biomedical hypothesis generation system. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1633–1642, New York, NY, USA, 2017. ACM.

[40] Justin Sybrandt, Micheal Shtutman, and Ilya Safro. Large-scale validation of hypothesis generation systems via candidate ranking. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1494–1503, 2018.

[41] Richard Van Noorden. Global scientific output doubles every nine years. *Nature news blog*, 2014.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[43] Marc Weeber, Henny Klein, Alan R Aronson, James G Mork, LT De Jong-van Den Berg, and Rein Vos. Text-based discovery in biomedicine: the architecture of the dad-system. In *Proceedings of the AMIA Symposium*, page 903. American Medical Informatics Association, 2000.

[44] Stephen Wilson, Angela Dawn Wilkins, Matthew V Holt, Byung Kwon Choi, Daniel Konecki, Chih-Hsu Lin, Amanda Koire, Yue Chen, Seon-Young Kim, Yi Wang, et al. Automated literature mining and hypothesis generation through a network of medical subject headings. *BioRxiv*, page 403667, 2018.

[45] Meliha Yetisgen-Yildiz and Wanda Pratt. Evaluation of literature-based discovery systems. In *Literature-based discovery*, pages 101–113. Springer, 2008.

[46] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 1(5), 2019.

Reproducability Details

Training Graph Embedding When optimizing our semantic graph embedding, we find that maximal performance is achieved using a compute cluster of twenty twenty-four-core machines. Within the 72h time restriction of the Palmetto super computing cluster, we have enough time to see every edge in the graph 10 times, in the case of the 256-dim embedding, and 5 times in the case of the 512-dim embedding. Once complete, we are ready to begin training the AGATHA deep learning hypothesis generation model.

Training Ranking Model We minimize the ranking loss over all published predicates using the LAMB optimizer [46]. This allows us to efficiently train using very large batch sizes, which is necessary as we leverage 10 NVIDIA V100 GPUs to effectively process 600 positive samples (and therefore 2,400 total samples) per batch. In terms of hyperparameters, we select a learning rate of $\eta = 0.01$ with a linear warm up of 1,000 batches, a margin of $m = 0.1$, a neighborhood sub-sampling rate of $s = 15$, and we perform cross-validation on a 1% random holdout to provide early stopping and to select the best model with respect to validation loss. Due to the large size of training data, one epoch consists of only 10% of the overall training data. This process is made easier through the helpful Pytorch-Lightning library [12].

Layer Name	Input Dim.	Output Dim.	Num. Params
Linear (ReLU)	512	512	262656
Enc. M.H.Att.	512	512	1050624
Enc. Dropout(0.1)	512	512	0
Enc. LayerNorm	512	512	1024
Enc. Linear (ReLU)	512	1024	524800
Enc. Dropout(0.1)	512	512	0
Enc. Linear (ReLU)	1024	512	525312
Enc. Dropout(0.1)	512	512	0
Enc. LayerNorm	512	512	1024
Encoder 2	512	512	2102272
Encoder 3	512	512	2102272
Encoder 4	512	512	2102272
Linear (sigmoid)	512	1	513

Table 3: Layers and parameter counts for the AGATHA transformer model.

Node Type	Count
Sentence	140,913,505
Predicate	19,268,319
Lemma	12,718,832
Entity	10,240,635
Coded Term	488,923
n-Grams	333,862
Total Nodes	183,964,076
Total Edges	12,362,325,167

Table 4: Graph Size of 2015 Validation Dataset