AGATHA: Automatic Graph mining And Transformer based Hypothesis generation Approach

Justin Sybrandt*
School of Computing
Clemson University
jsybran@clemson.edu

Ilya Tyagin
School of Computing
Clemson University
ityagin@clemson.edu

Michael Shtutman
Drug Discovery and
Biomedical Sciences
University of S. Carolina
shtutmanm@sccp.sc.edu

Ilya Safro
School of Computing
Clemson University
isafro@clemson.edu

ABSTRACT

Medical research is risky and expensive. Drug discovery requires researchers to efficiently winnow thousands of potential targets to a small candidate set. However, scientists spend significant time and money long before seeing the intermediate results that ultimately determine this smaller set. 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 learns a data-driven ranking criteria to recommend new biomedical connections. We massively validate our system with a temporal hold-out wherein we predict connections first introduced after 2015 using data published beforehand. We additionally explore biomedical sub-domains, and demonstrate AGATHA's predictive capacity across the twenty most popular relationship types. Furthermore, we perform an ablation study to examine the aspects of our semantic network that most contribute to recommendation quality. Overall, AGATHA achieves best-in-class recommendation quality when compared to other hypothesis generation systems built to predict across all available biomedical literature. **Reproducibility:** All code, experimental data, and pre-trained models are available online: sybrandt.com/2020/agatha .

CCS CONCEPTS

• **Applied computing** → *Bioinformatics*; *Document management and text processing*; • **Computing methodologies** → *Learning latent representations*; *Neural networks*; **Information extraction**; **Semantic networks**.

KEYWORDS

Hypothesis Generation, Literature-Based Discovery, Transformer Models, Semantic Networks, Biomedical Recommendation,

*Now with Google Brain. Contact: jsybrandt@google.com.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412684

ACM Reference Format:
Justin Sybrandt, Ilya Tyagin, Michael Shtutman, and Ilya Safro. 2020. AGATHA: Automatic Graph mining And Transformer based Hypothesis generation Approach . In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412684

1 INTRODUCTION

As the rate of global scientific output continues to climb [39], 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]. 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 [35]. Hypothesis generation systems aim to exploit the wealth of public scientific text by automatically identifying plausible new research directions [32, 35]. However, existing systems are often either specialized to particular biomedical subdomains [25, 42], or require significant human interpretation [33, 34, 37].

This work presents AGATHA, a deep learning technique to automatically identify plausible biomedical hypotheses across the entire span of biomedical literature. In this context, a hypothesis is a proposed association between two entities of interest. AGATHA constructs and embeds a large semantic graph containing over ten-billion edges from the literature, and then trains a transformer encoder [40] to rank plausible connections. This work *fundamentally changes* our development of hypothesis generation systems that are in active use for drug discovery. Our initial system, MOLIERE [37], constructs a smaller semantic network, which it uses to identify key abstracts and perform topic modeling based on user-supplied connections of interest. Using heuristically derived ranking criteria [38], the MOLIERE system successfully identified a novel gene treatment target for HIV-associated Neurodegenerative Disorder through the inhibition of DDX3X, which was confirmed in wet lab experiments [3].

AGATHA establishes a next-generation approach to hypothesis generation by challenging many of the assumptions that underpin the MOLIERE system. Most dramatically, AGATHA replaces MOLIERE's heuristically derived ranking criteria with a data-driven measure that we learn directly from the distribution of existing biomedical connections. We observe substantial performance improvements when comparing MOLIERE and AGATHA ranking

quality. Specifically, on the same benchmark with the same training data, AGATHA scores a ROC AUC of 0.97 and a PR AUC of 0.98, whereas MOLIERE scores 0.72 and 0.82 respectively [36]. Furthermore, AGATHA performs queries orders of magnitude faster than MOLIERE, and requires less human oversight than many contemporary systems [9, 33, 37]. Specifically, AGATHA can perform nearly 500 queries per second on one GPU, while MOLIERE required an average of 100 seconds per query [36]. For these reasons, we currently supply an up-to-date instance of AGATHA for use in the scientific community, and with help from the NSF we are in the process of releasing an update specialized for accelerating COVID-19 research.

In addition to increased quality and performance compared to MOLIERE, AGATHA has advantages when compared to other systems in the state of the art. Many are constructed to predict implicit connections from domain-specific graphs [25, 42]. In contrast, AGATHA is trained to predict connections from the entire scope of biomedical literature. Yet still, in this work we find that our more generalized ranking model is capable of high quality domain-specific connection recommendation. Other modern systems require human oversight, for instance via the interpretation of visualizations or clusters [9, 33, 37]. While these techniques provide much needed intractability to the hypothesis generation pipeline, the human-in-the-loop strategy inevitably slows discovery and introduces additional bias. As a result of these limitations inherent to many modern techniques, many provide very restricted validation experiments, often consisting of only a handful of queries [15, 20, 21, 29]. By combining automated analysis, a more generalized prediction space, and high-performance queries, we are able to validate AGATHA on thousands of queries, and perform large scale recommendation across a wide set of possibilities.

Deployment Details: AGATHA supports collaborations with the Drug Discovery and Biomedical Sciences department at the University of South Carolina, as well as other departments within Clemson. We are working with the startup Scifeat to produce an interpretable dashboard for AGATHA results. Using this interface, we are creating a large dataset of generated hypotheses pertraining to COVID-19 for use by the broader biomedical community. For organizations wishing to work with AGATHA directly, the entire system is open source and we provide models trained on up-to-date datasets.

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 [27]. (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] (over 10 billion edges) in a common space using the PyTorch-BigGraph embedding [24]. (3) We perform an ablation study to compare the quality of the AGATHA ranking criteria when trained against various subgraphs of our full semantic network. (4) We validate our system using the massive validation techniques presented in [38], and also demonstrate the ability of AGATHA to generalize across biomedical subdomains. We compare these results to a similar ranking model proposed by Edge2Vec [13], which outperformed other methods.

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 . We additionally encourage enthusiastic readers to view a more detailed long version of this paper online[1].

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 [35].

Our former approach to address these challenges is posed by the MOLIERE system [37], and its accompanying plausibility ranking criteria [38]. 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].

PyTorch-BigGraph (PTBG) [24] is an open-source, large-scale, distributed graph-embedding technique aimed at heterogeneous information networks [30]. 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 [40] 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

[1] Long form version: https://arxiv.org/abs/2002.05635

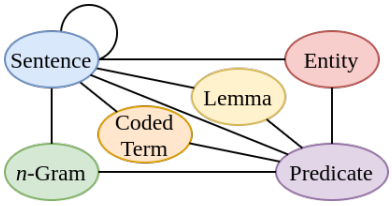

Figure 1: AGATHA multi-layered graph schema.

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:

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

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, to convert raw-text sources into a semantic graph (Fig. 1). An embedding of this graph enables our learned ranking criteria.

Text Pre-Processing. We begin with raw MEDLINE XML files [2]. 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. *For fair validation of our system, we additionally discard any document (or another information) 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 [27]. 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 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 [3] 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 $O(n^2 d)$) 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 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

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

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

complexity for identifying all nearest-neighbors from $O(dn^2)$ to $O\left(qdn^2/32bpm\right)$.

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 [24]. 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 $x \rightarrow y$ and $y \rightarrow x$.

When computing 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 is a predetermined embedding dimensionality. In our experiments we consider $N = 512$. 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:

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

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:

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

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 [38]. 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.

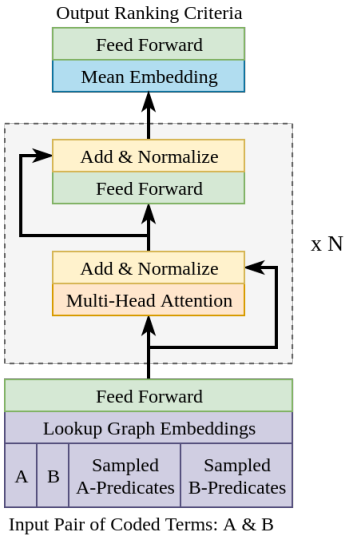

Figure 2: AGATHA ranking transformer encoder.

4 RANKING PLAUSIBLE CONNECTIONS

We train a model to rank published SemRep [4] predicates above noisy negative samples using the transformer architecture [40]. 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:

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

Negative Samples We cannot learn to rank positive training examples in isolation. Instead, we first generate negative samples to accompany each published predicate. This includes 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 ranking formulation allows the model to be certain of some hypotheses, and shaky on others without facing a penalty, which is important for real-world applications. 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 [43]. Without the resource to perform expensive wetlab experiments, most designing hypothesis generation systems evaluate their system on the ability to uncover recent findings using historical data [38]. An additional challenge comes from the broad scope of AGATHA. While most hypothesis generation systems focus on particular biomedical subdomains [25, 32, 42], this system incorporates the entire MEDLINE dataset. Therefore, we compare our proposed system to both our prior work, MOLIERE [37], as well as a biomedically-specialized knowledge graph embedding technique intended for hypothesis generation applications, Edge2Vec [13].

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 [38]. Specifically, the MOLIERE system presents experimental numbers for various training-data scenarios for the same 2015 temporal holdout as used in this work [36]. 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.

Edge2Vec Comparison. We compare AGATHA performance to the recent ranking model built using Edge2Vec embeddings [13]. Edge2Vec is a biomedical knowledge graph embedding technique intended for a similar applications as AGATHA, which we summarize in Section 7. This method was also chosen because it significantly outperforms several other methods (see [13]) which makes our comparison with them meaningless. We train Edge2Vec on the knowledge graph of predicate statements, wherein edges are triples of the form "subject-verb-object." We train a model that learns to rank a potential subject-object connection by concatenating their corresponding Edge2Vec representations and then projecting down to the unit interval. We fit this model with the same margin ranking loss objective and optimizer as used to train AGATHA.

System Instance	ROC AUC	PR AUC
AGATHA	**0.97**	**0.98**
Edge2Vec	0.92	0.92
MOLIERE [Abstract]	0.72	0.82
MOLIERE [Full Text]	0.80	0.78

Table 1: Benchmark comparison between MOLIERE and AGATHA on the same benchmark.

Ablation Study. To further explore the impact that each aspect of the AGATHA semantic graph has on its predictive capacity, we perform an ablation study. In this study, we selectively remove different aspects of the semantic graph, retrain graph embeddings, and then retrain the AGATHA ranking model. We then evaluate the result against the same queries described above for subdomain recommendation. We consider the following ablations: the entire graph (Full), the graph with sentence nodes removed (No Sent.), with entity nodes removed (No Ent.), with lemma nodes removed (No Lemmas), with *n*-Gram nodes removed (No *n*-Grams), with the sentence nearest-neighbors edges removed (No Sent. *k*-NN), and lastly, the smallest graph that still enables training, containing only predicates and coded terms (Only Pred.).

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.

6 RESULTS

First, we compare the performance of AGATHA to MOLIERE using the benchmark established in [36]. These results can be found in Table 1. Note that we include results for the version of MOLIERE trained on the same dataset as AGATHA, as well as the higher-performing version trained on full text papers [36], and also the Edge2Vec model trained on the predicate knowledge network. We observe that AGATHA significantly outperforms MOLIERE. This is because the AGATHA ranking criteria is learned directly from the distribution of terms, as opposed to the heuristic criteria used in MOLIERE [38]. Furthermore, we find that AGATHA outperforms Edge2Vec. AGATHA leverages significantly more information when learning embeddings, and receives additional information in the form of samples neighbors when performing predictions.

Beyond quality, the find that AGATHA performs queries faster than MOLIERE, which has to perform expensive graph traversals and topic modeling for each query. In contrast, AGATHA can store all needed embeddings in memory, and can batch process queries on GPU. While MOLIERE spends an average of 100 seconds per query, and the full-text version spends over 75 minutes [36], *AGATHA can perform around 500 queries per second.*

Next, we compare AGATHA performance across popular biomedical subdomains in Table 2. This task requires AGATHA to recommend new research directions from large many-to-many collections of queries. As a result, the area under the precision-recall curve (PR AUC) is likely the most meaningful single metric, followed by the precision (P) and average precision (AP). We observe that across many subdomains, AGATHA outperforms the baseline established by Edge2Vec. For instance, we find that the average PR AUC across trials is higher (0.212 compared to 0.191) and that AGATHA has a higher top-10 precision (0.360 compared to 0.335). Looking at specific subdomains, we find that AGATHA is best able to recommend gene to cell-function (gngm:cell) and gene to neoplastic process (gngm:neop).

Lastly, we compare the ablations of AGATHA in Table 3. Here we present the average performance across the subdomain recommendation benchmark for each model. Interestingly, we find that the AGATHA model that removes *n*-Grams outperforms the larger model. We found, while developing MOLIERE, that *n*-grams were crucial for expert analysis of topic modeling results, and included them within AGATHA as a way to facilitate a similar alternate query strategy. However, we find that the degree distribution of our *n*-grams is significantly different from the other textual node types. As a result, there appears to be an unexpected convergence issue when performing graph embeddings, which contributes to decreases recommendation quality. Beyond this single unexpected result, we also find that there are substantial performance benefits to including the sentence subgraph, as removing sentence nodes or nearest-neighbors edges has the two largest impacts in performance. This finding confirms our initial assumption that keyword associations alone (provided by the lemmas and entities) are insufficient to accurately identify semantically similar sentences. With high-quality sentence embeddings, it would appear that we also receive higher quality predicate embeddings, which are directly used by the AGATHA ranking model.

7 RELATED WORK

Foster et al. [12] 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 [17, 19, 41], or predicting links with a graph of biomedical entities [11, 28]. 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 [34]. 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 [22]. However, ABC-based techniques have significant limitations [31], including their similarity metrics defined

	ROC AUC		PR AUC		RR		P@10		P@100		AP@10		AP@100	
	A	E	A	E	A	E	A	E	A	E	A	E	A	E
aapp:dsyn	0.77	0.70	0.23	0.19	0.20	1.00	0.30	0.40	0.32	0.27	0.32	0.54	0.37	0.33
aapp:aapp	0.77	0.72	0.10	0.09	0.07	0.25	0.00	0.10	0.12	0.10	0.00	0.25	0.13	0.12
aapp:cell	0.72	0.72	0.23	0.20	0.33	1.00	0.60	0.20	0.31	0.23	0.51	0.60	0.39	0.27
aapp:gngm	0.74	0.68	0.24	0.19	0.20	1.00	0.20	0.40	0.43	0.29	0.23	0.73	0.41	0.39
aapp:neop	0.73	0.72	0.28	0.26	1.00	1.00	0.60	0.20	0.40	0.34	0.57	0.83	0.49	0.39
bacs:aapp	0.77	0.66	0.15	0.11	0.50	0.25	0.30	0.30	0.23	0.17	0.53	0.34	0.26	0.22
bacs:gngm	0.74	0.64	0.19	0.16	1.00	0.50	0.30	0.50	0.25	0.23	0.87	0.71	0.36	0.45
bpoc:aapp	0.76	0.69	0.21	0.17	0.50	0.12	0.40	0.30	0.32	0.30	0.59	0.22	0.38	0.32
cell:aapp	0.72	0.71	0.19	0.23	0.50	1.00	0.40	0.60	0.25	0.30	0.57	0.59	0.33	0.52
dsyn:dsyn	0.78	0.75	0.12	0.17	0.25	0.50	0.10	0.60	0.18	0.28	0.25	0.72	0.21	0.48
dsyn:humn	0.77	0.77	0.19	0.18	0.50	0.12	0.30	0.10	0.25	0.23	0.48	0.12	0.32	0.20
gngm:aapp	0.74	0.70	0.22	0.17	1.00	1.00	0.40	0.10	0.37	0.19	0.58	1.00	0.41	0.20
gngm:celf	0.69	0.74	0.35	0.38	0.33	0.25	0.30	0.40	0.49	0.45	0.32	0.31	0.43	0.45
gngm:cell	0.75	0.67	0.24	0.19	1.00	1.00	0.60	0.30	0.35	0.30	0.80	0.77	0.46	0.40
gngm:dsyn	0.77	0.69	0.17	0.15	0.50	1.00	0.30	0.30	0.27	0.28	0.56	0.87	0.32	0.41
gngm:gngm	0.78	0.71	0.18	0.13	0.14	0.20	0.30	0.20	0.23	0.12	0.22	0.21	0.23	0.15
gngm:neop	0.73	0.75	0.36	0.38	1.00	1.00	0.60	0.80	0.46	0.47	0.81	0.88	0.53	0.65
orch:gngm	0.77	0.72	0.22	0.18	0.50	1.00	0.20	0.30	0.28	0.23	0.42	0.81	0.30	0.32
phsu:dsyn	0.77	0.71	0.20	0.17	1.00	0.50	0.50	0.30	0.29	0.29	0.55	0.36	0.40	0.33
topp:dsyn	0.77	0.72	0.16	0.12	0.50	0.33	0.50	0.30	0.26	0.23	0.57	0.32	0.41	0.23
Mean	**0.752**	0.709	**0.212**	0.191	0.551	**0.651**	**0.360**	0.335	**0.303**	0.265	0.488	**0.559**	**0.357**	0.342

Table 2: The above lists metrics that quantify recommendation quality across popular biomedical subdomains. We compare Agatha (A) with Edge2Vec (E).

	Full	Only Pred.	No Sent.	No Sent. k-NN	No Ent.	No Lemmas	No n-Grams
ROC	0.745	0.738	0.677	0.734	0.740	0.738	**0.752**
PR	0.205	0.208	0.148	0.192	0.195	0.199	**0.211**
RR	0.512	0.562	0.483	0.526	**0.612**	0.449	0.551
P@10	0.315	0.355	0.180	0.260	0.320	0.325	**0.360**
P@100	0.288	0.287	0.169	0.259	0.249	0.271	**0.303**
AP@10	0.423	0.486	0.422	0.474	0.486	0.421	**0.487**
AP@100	0.346	0.352	0.213	0.307	0.321	0.325	**0.357**

Table 3: Average performance of the AGATHA across ablations. Reported values represent the average of all subdomain results.

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 [23].

A completely different strategy of performing LBD is proposed by Spangler et al. in [33]. 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 [26] 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. [29] 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) The GrEDeL LSTM model is trained to only discover drug-disease associations, and does not generalize to other biomedical subdomains. (3) GrEDeL relies on 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.

Edge2Vec [13] is a biomedically-specialized knowledge graph embedding that uses the well-proven approach of aggregating graph random walks. This approach differs from more common walk strategies by utilizing an estimated edge-type transition matrix to capture the context of nodes, and is paired with the Expectation-Maximization model to train embeddings. However, we observe

that the Edge2Vec model, which uses a common one-layer neural network strategy to fit embeddings, is limited in its ability to preserve higher-order graph features. Importantly, the Edge2Vec model is intended as a knowledge graph embedding strategy, and therefore can be augmented in a number of ways to facilitate various hypothesis generation applications.

ACKNOWLEDGMENTS

We would like to thank the CCIT staff who manage the Palmetto Supercomputer at Clemson University, where we ran all of our experiments. This work was additionally supported by NSF #1633608.

REFERENCES

[1] [n. d.]. Citations Added to MEDLINE by Fiscal Year. https://www.nlm.nih.gov/bsd/stats/cit_added.html
[2] [n. d.]. Semantic Types. https://mmtx.nlm.nih.gov/MMTx/semanticTypes.shtml
[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. 2019. Inhibition of the Dead Box RNA Helicase 3 prevents HIV-1 Tat and cocaine-induced neurotoxicity by targeting microglia activation. *Journal of Neuroimmune Pharmacology* (2019), 1–15.
[4] Patrick Arnold and Erhard Rahm. 2015. SemRep: A repository for semantic mapping. *Datenbanksysteme für Business, Technologie und Web* (2015).
[5] Nadine Bakkar, Tina Kovalik, Ileana Lorenzini, Scott Spangler, Alix Lacoste, Kyle Sponaugle, Philip Ferrante, Elenee Argentinis, Rita Sattler, and Robert Bowser. 2018. 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 (2018), 227–247.
[6] Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676* (2019).
[7] D. Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (2012), 77–84.
[8] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
[9] Ying Chen, JD Elenee Argentinis, and Griff Weber. 2016. IBM Watson: how cognitive computing can be applied to big data challenges in life sciences research. *Clinical therapeutics* 38, 4 (2016), 688–701.
[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. 2018. Literature-based automated discovery of tumor suppressor p53 phosphorylation and inhibition by NEK2. *Proceedings of the National Academy of Sciences* 115, 42 (2018), 10666–10671.
[11] Lauri Eronen and Hannu Toivonen. 2012. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC bioinformatics* 13, 1 (2012), 119.
[12] Jacob G Foster, Andrey Rzhetsky, and James A Evans. 2015. Tradition and innovation in scientists' research strategies. *American Sociological Review* 80, 5 (2015), 875–908.
[13] Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, Jeremy Yang, Christopher Gessner, Brian Foote, David Wild, Ying Ding, et al. 2019. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. *BMC bioinformatics* 20, 1 (2019), 306.
[14] Vishrawas Gopalakrishnan, Kishlay Jha, Guangxu Xun, Hung Q Ngo, and Aidong Zhang. 2018. Towards self-learning based hypotheses generation in biomedical text domain. *Bioinformatics* 34, 12 (2018), 2103–2115.
[15] Vishrawas Gopalakrishnan, Kishlay Jha, Aidong Zhang, and Wei Jin. 2016. 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*. 23–30.
[16] Sam Henry. 2019. Indirect Relatedness, Evaluation, and Visualization for Literature Based Discovery. (2019).
[17] Dimitar Hristovski, Carol Friedman, Thomas C Rindflesch, and Borut Peterlin. 2006. Exploiting semantic relations for literature-based discovery. In *AMIA annual symposium proceedings*, Vol. 2006. 349.
[18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 117–128.
[19] Rob Jelier, Martijn J Schuemie, Antoine Veldhoven, Lambert CJ Dorssers, Guido Jenster, and Jan A Kors. 2008. Anni 2.0: a multipurpose text-mining tool for the life sciences. *Genome biology* 9, 6 (2008), R96.
[20] Kishlay Jha, Guangxu Xun, Yaqing Wang, Vishrawas Gopalakrishnan, and Aidong Zhang. 2018. Concepts-bridges: Uncovering conceptual bridges based on biomedical concept evolution. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1599–1607.
[21] Kishlay Jha, Guangxu Xun, Yaqing Wang, and Aidong Zhang. 2019. 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*. 843–851.
[22] Yong Hwan Kim and Min Song. 2019. A context-based ABC model for literature-based discovery. *PloS one* 14, 4 (2019).
[23] 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. 2009. Literature-related discovery. *Annual review of information science and technology* 43, 1 (2009), 1–71.
[24] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference*.
[25] Anthony ML Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. 2011. BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome biology* 12, 6 (2011), R57.
[26] 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. 2014. DiseaseConnect: a comprehensive web server for mechanism-based disease–disease connections. *Nucleic acids research* 42, W1 (2014), W137–W146.
[27] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669* (2019).
[28] Murali K Pusala, Ryan G Benton, Vijay V Raghavan, and Raju N Gottumukkala. 2017. Supervised approach to rank predicted links using interestingness measures. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 1085–1092.
[29] Shengtian Sang, Zhihao Yang, Xiaoxia Liu, Lei Wang, Hongfei Lin, Jian Wang, and Michel Dumontier. 2018. GrEDeL: A knowledge graph embedding based method for drug discovery from biomedical literatures. *IEEE Access* 7 (2018), 8404–8415.
[30] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 17–37.
[31] Neil R Smalheiser. 2012. Literature-based discovery: Beyond the ABCs. *Journal of the American Society for Information Science and Technology* 63, 2 (2012), 218–224.
[32] Scott Spangler. 2015. *Accelerating Discovery: Mining Unstructured Information for Hypothesis Generation*. Chapman and Hall/CRC.
[33] 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. 2014. Automated hypothesis generation based on mining scientific literature. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1877–1886.
[34] Don R Swanson. 1986. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine* 30, 1 (1986), 7–18.
[35] Don R Swanson. 1986. Undiscovered public knowledge. *The Library Quarterly* 56, 2 (1986), 103–118.
[36] Justin Sybrandt, Angelo Carrabba, Alexander Herzog, and Ilya Safro. 2018. Are Abstracts Enough for Hypothesis Generation?. In *2018 IEEE International Conference on Big Data (Big Data)*. 1504–1513. https://doi.org/10.1109/bigdata.2018.8621974
[37] Justin Sybrandt, Michael Shtutman, and Ilya Safro. 2017. MOLIERE: Automatic Biomedical Hypothesis Generation System. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1633–1642. https://doi.org/10.1145/3097983.3098057
[38] Justin Sybrandt, Micheal Shtutman, and Ilya Safro. 2018. Large-Scale Validation of Hypothesis Generation Systems via Candidate Ranking. In *2018 IEEE International Conference on Big Data (Big Data)*. 1494–1503. https://doi.org/10.1109/bigdata.2018.8622637
[39] Richard Van Noorden. 2014. Global scientific output doubles every nine years. *Nature news blog* (2014).
[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[41] Marc Weeber, Henny Klein, Alan R Aronson, James G Mork, LT De Jong-van Den Berg, and Rein Vos. 2000. Text-based discovery in biomedicine: the architecture of the DAD-system.. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 903.
[42] 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. 2018. Automated literature mining and hypothesis generation through a network of Medical Subject Headings. *BioRxiv* (2018), 403667.
[43] Meliha Yetisgen-Yildiz and Wanda Pratt. 2008. Evaluation of literature-based discovery systems. In *Literature-based discovery*. Springer, 101–113.