Learning to Rank Articles for Molecular Queries

Galia Nordon[1*], Aviram Magen[1*], Ido Guy[2], Kira Radinsky[1]

[1]Technion-Israel institute of technology
[2]eBay
galiasn@gmail.com, avirammagen11@gmail.com, iguy@ebay.com, kirar@cs.technion.ac.il

Abstract

The cost of developing new drugs is estimated at billions of dollars per year. Identification of new molecules for drugs involves scanning existing bio-medical literature for relevant information. As the potential drug molecule is novel, retrieval of relevant information using a simple direct search is less likely to be productive. Identifying relevant papers is therefore a more complex and challenging task, which requires searching for information on molecules with similar characteristics to the novel drug. In this paper, we present the novel task of ranking documents based on novel molecule queries. Given a chemical molecular structure, we wish to rank medical papers that will contribute to a researcher's understanding of the novel molecule drug potential. We present a set of ranking algorithms and molecular embeddings to address the task. An extensive evaluation of the algorithms is performed over the molecular embeddings, studying their performance on a benchmark retrieval corpus, which we share with the community. Additionally, we introduce a heterogeneous edge-labeled graph embedding approach to address the molecule ranking task. Our evaluation shows that the proposed embedding model can significantly improve molecule ranking methods. The system is currently deployed in a targeted drug delivery and personalized medicine research laboratory.

Introduction

Drug discovery is the process of identifying new medicines. Once a drug candidate is identified, it is optimized to increase affinity, efficacy, and stability, while reducing side effects. A compound that passes the optimization will then move to drug development process, prior to clinical trials. The estimated cost of the process is 2 billion dollars and it may last between 10 and 15 years, while only 0.1% of the drugs that enter pre-clinical testing progress to human testing, and only 20% of these are approved by the FDA (Suresh and Basu 2008). One of the first steps of the drug discovery process is the generation of candidate molecular compounds. During this process, prior trials and publication regarding similar substances are reviewed in order to ensure the novelty of the compound and evaluate its characteristics. In this work, we present a similarity search algorithmic framework for drug discovery that can assist in this process. Given a novel chemical molecular structure, which has not been previously developed, we wish to identify and rank medical papers most relevant to the molecule. For example, suppose the year is 1970 and a researcher is developing Captopril, our query will be a molecule; in this example the molecule is described using SMILES (Weininger 1988) string representation and the result should be a list of relevant articles to this molecule. For example, papers discussing drugs or molecules with similar toxicity, effect etc.

We explore two types of embeddings for our query: structural embedding, which relates to the chemical structure of the molecule and linguistic embedding, based on the contexts of the molecule in a large PubMed corpus. We present a novel methodology to combine the two approaches by constructing a graph G, with several types of nodes, representing text documents, structural molecular fingerprints, and molecules. The graph's edges represent the similarity between the different nodes. We leverage a graph embedding algorithm to produce node representations using random walks. This enables us to jointly learn representations of both documents and molecules.

The contribution of this work is threefold: (1) We present the novel task of learning to rank documents for newly developed molecules. To the best of our knowledge, we are the first to address the problem of ranking medical papers for a novel molecule; (2) We study several different ranking algorithms for this task and present comparative empirical results; (3) We study several embedding approaches to represent the molecules and documents for this problem. Specifically, we present a novel graph-based embedding algorithm that represents both documents and molecules in the same space. Empirical results show this approach yields superior results for the task across all ranking algorithms.

Related Work

With the development of deep learning, convolutional neural networks, initially developed for image recognition, were successfully applied on molecular graphs (Coley et al. 2017) and on 2D depictions of molecules (Goh et al. 2017). At the same time, natural language processing techniques were adopted to learn from classical molecule representations, e.g., molecular fingerprints (Wan and Zeng 2016), SMILES (Olivecrona et al. 2017), and graph representations

*These authors contributed equally.

of compounds (Kearnes et al. 2016). In addition, the information retrieval TF-IDF method was applied to ECFP for the prediction of compound proteins (Wan and Zeng 2016) and LDA was used for the modeling of chemical topics (Schneider et al. 2017). Deep generative models opened up new opportunities to leverage molecular embedding for unsupervised tasks, such as the generation of molecules and the discovery of drugs (Gómez-Bombarelli et al. 2018; Kadurin et al. 2017; Harel and Radinsky 2018).Simple networks have been used for discovering novel associations in medical and biomedical papers (Swanson and Smalheiser 1999; Spangler et al. 2014). Spangler et al. apply text mining techniques to identify entities and relations relevant to a specific query. Although numerous studies have been conducted in the broad field, to the best of our knowledge, we are the first to present the task of ranking documents given a novel query.

We construct a graph representing documents, molecules, and molecular fingerprints and study several embedding methods for our task. Network embedding has been shown helpful in many applications, such as node classification (Bhagat, Cormode, and Muthukrishnan 2011) and link prediction (Liben-Nowell and Kleinberg 2007).

Encoder-decoder models have been proposed to address network representation problems. To scale for large graphs, several random-walk based approaches have been proposed. The resulting walks generate candidate paths in the graph, which are then used in a word2vec-based network representation learning frameworks, such as DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), TADW (Cao, Lu, and Xu 2015), LINE (Tang et al. 2015), and node2vec (Grover and Leskovec 2016). The skip-gram model (Mikolov et al. 2013), used in these frameworks, aims to train a model based on the context of each node.Thus Deepwalk, LINE, and node2vec are not effective for representing heterogeneous networks, where some nodes are over-represented. Intuitively, candidate paths will contain more nodes from the over-represented group, thus embeddings for under-represented groups will be less effective. In this work, for example, text document nodes are over-represented as compared to the chemical fingerprints. We present an embedding that will prevent such bias and show empirical results for the approach superiority as compared to the above state-of-the-art approaches.

Molecule and Document Representation

In this section, we explore learning a representation that maps both text documents and chemical structures into a single embedding space. Capturing the chemical structure along with the text allows, in turn, to better rank documents for a new molecular query given as input to the search framework.

The Molecule-Document Graph

We construct a weighted graph G, with three types of nodes: (1) historically known molecules, M (2) molecular fingerprints[1], F, and (3) medical publications, D. An illustration of the graph is given in Figure 1. We define several types of edges R_i and their corresponding weights w_i, as follows:

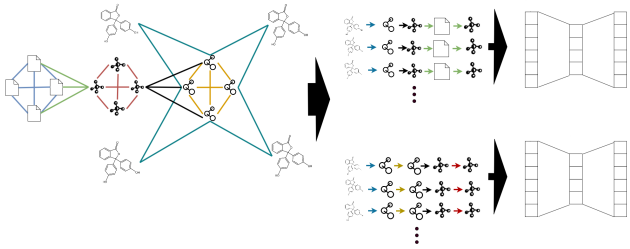

Figure 1: Pipeline of molgraph2vec methods. The heterogeneous network (left). Random paths generated using meta-path constraints (middle). Skip gram model for learning node embedding (right).

Molecule Relation. $R_{m_1 \to m_2}$: the edge's weight is defined as $weight_{m_1 \to m_2} = sim(m1, m2)$, where sim is a similarity function between the word representations of the known drugs $m_1 \in M$ and $m_2 \in M$ as appears in articles. In our experiments, we use cosine similarity for sim and word2vec for text representation of $m1$ and $m2$ (Arora, Liang, and Ma 2017).

Document Relation. $R_{d_1 \to d_2}$: the weight of the edge is defined as $weight_{d_1 \to d_2} = sim(d_1, d_2)$, where sim is a similarity function between the text representations of the paper d_1 and the paper d_2. For $d1$ and $d2$ we used word2vec for text representation (Arora, Liang, and Ma 2017) and cosine similarity as the similarity function.

Fingerprint Relation. $R_{f_1 \to f_2}$: the weight of the edge is defined as $weight_{f_1 \to f_2} = sim(f_1, f_2)$, where sim is a similarity function between the chemical representations of the fingerprint $f_1 \in F$ and the fingerprint $f_2 \in F$. We use cosine similarity as the similarity function and Morgan FP embeddings as the embeddings of f_i.

Molecule-Fingerprint Relation. $R_{m \to f}$: the weight of the edge is defined as $weight_{m \to f} = sim(m, f)$, where sim is a similarity function. Following the similarity function suggested by (Jaeger, Fulle, and Turk 2018), we consider the number of times that the fingerprint $f \in F$ appears in the molecule $m \in M$ as the weight.

Molecule-Document Relation. $R_{m \to d}$: the weight of the edge is defined as $weight_{m \to d} = sim(m, d)$, where sim is a similarity function between the text representations of the drug $m \in M$ and the paper $d \in D$. We use cosine similarity as the similarity function and weighted word2vec for text representation.

Heterogeneous Graph Embedding

The molecule-document graph poses a challenge, as it represents a weighted heterogeneous network with multiple types of nodes, in our case molecules, fingerprints, and documents. Since the molecule-document graph is a highly unbalanced graph, consisting of a high number of documents

[1]Molecular fingerprints are a widely accepted way to represent molecules for similarity related tasks. In our work we use ECFPs or Morgan fingerprints (Rogers and Hahn 2010).

and a significantly lower number of potential fingerprints, classic random walks might lead to a bias towards paths through document nodes only. Such paths might create in turn biased embeddings that will not map well documents and molecules into the same space, which is essential for our task. A similar issue was observed in other heterogeneous networks as shown by (Sun et al. 2011).

Metapath2vec (Dong, Chawla, and Swami 2017) was recently proposed for creating graph embeddings for heterogeneous networks. It extends the basic node2vec approach with metapath-based random walks. That is, the random walk is constrained by a metapath. Formally, the random walk between a node v_1 and v_n for a metapath $p = R_1, \ldots, R_n$, where R_i is a label type as defined in the previous section, might only take the form: $v_1 \xrightarrow{R_1} v_2 \xrightarrow{R_2} \ldots \xrightarrow{R_n} v_n$. The approach attempts to preserve both the structure and semantics of a given heterogeneous network. The random walks allow creating embeddings based on the heterogeneous neighborhood of a node, which are then leveraged using the classic skip-gram model for the node embeddings. The method was shown to improve performance for heterogeneous graphs.

Metapath2vec requires selecting a specific metapath scheme $p = R_1, \ldots, R_n$ to guide the random walks. For example, (Dong, Chawla, and Swami 2017) surveyed qualitative metapath-based prior work and identified that for heterogeneous academic networks the most efficient metapaths are author-paper-author and author-paper-venue-paper-author paths. However, no prior work has been done in the field of molecule and document graph representation. In the next section, we first extend the metapath2vec to support weighted heterogeneous networks. We then present a novel algorithm for heterogeneous networks when no metapath is defined apriori.

The Optimization Problem We define the edge-labeled heterogeneous graph feature learning problem as a maximum likelihood optimization problem. Our analysis is not limited to (un)directed graphs or (un)weighted graphs. Let $G = (V, E, L)$ be a given edge-labeled heterogeneous graph and let $f : V \rightarrow R^d$ be the desired function from nodes to the feature representations we wish to learn. For each node $u \in V$, we define the neighbourhood of type l, NbL, as the set of nodes that are connected to u via an edge of label l. Formally: $NbL(u, l) \quad \& := \{v \in V \mid (u, v) \in E \wedge label((u, v)) = l\}$.

Similar to metapath2vec, we extend the design of the skip-gram model for edge-labeled graphs. Our optimization objective function aims to maximize the log-probability of observing a graph neighborhood $NbL(u, l)$ for a node u conditioned both on its feature representation, given by f, and on the edge label l:

$$\max_f \Sigma_{u \in v, l \in L} \log Pr(NbL(u, l) \mid f(u), l) \qquad (1)$$

Two standard assumptions are made in order to make the optimization problem computationally efficient:

1. *Conditional independence*: We assume that the likelihood of observing a neighborhood node connected with an edge labeled as l is independent of observing any other neighborhood node connected with the same labeled edge given the feature representation of the source and a label $l \in L$.

   $\forall v_1, v_2 \in \mathrm{NbL}(u, l) : Pr(v_1 \mid f(u), l) \perp\!\!\!\perp Pr(v_2 \mid f(u), l)$

   We use this assumption in order to factorize the likelihood of observing the neighbours of u for a label l as:

   $$Pr(NbL(u, l) \mid f(u), l) = \Pi_{n_i \in NbL(u,l)} Pr(n_i \mid f(u))$$

2. *Symmetry in feature space*: We assume that a source node $u \in V$ and a neighborhood node $n_i \in V$ have a symmetric effect on each other in the feature space. Consequently, the conditional likelihood of transforming from u, given by its embedding under f, to a neighbourhood node n_i will be modeled as a softmax unit parametrized by the dot product features of each source-neighborhood node pair:

   $$Pr(n_i | f(u)) = \frac{\exp\left(f(n_i) \cdot f(u)\right)}{\Sigma_{v \in V} \exp\left(f(v) \cdot f(u)\right)}$$

The objective in Eq. 1, according to the above assumptions is therefore simplified to:

$$\begin{aligned} \max_f \Sigma_{u \in v, l \in L} \log\left(\Pi_{n_i \in NbL(u,l)} Pr(n_i | f(u))\right) = \\ \max_f \Sigma_{u \in v, l \in L} \Sigma_{n_i \in NbL(u,l)} \log Pr(n_i | f(u)) = \\ \max_f \Sigma_{u \in v, l \in L} \Big[-\log Z_u + \Sigma_{n_i \in NbL(u,l)} (f(n_i) \cdot f(u)) \Big] \end{aligned} \qquad (2)$$

As the partition function, $Z_u = \Sigma_{v \in V} \exp\left(f(v) \cdot f(u)\right)$, is expensive to compute for large graphs, we leverage negative sampling (Mikolov et al. 2013) to estimate the per-node partition. We optimize Eq. 2 using stochastic gradient ascent over the model parameters defining the features f.

We now describe how to estimate the neighbourhood NbL. Intuitively, word2vec leverages the skip-gram architecture to sample neighbours using a sliding window. Node2vec preforms randomized walks that sample many different neighborhoods of a given source node u. Each random walk creates a "sentence" of nodes where the context of a node is evaluated from its neighborhood. We use these random walks as input to the skip-gram architecture. We refer to the application of this method to the molecule-document graph as *molgraph2vec*.

Metapath Guided Random Walks As previously discussed, our network structure is heterogeneous, containing many different types of nodes, and therefore simple random walks tend to create biased embeddings (Dong, Chawla, and Swami 2017). We therefore suggest heterogeneous random walks. We define a metapath scheme $p = R_1, \ldots, R_n$, where R_i is a label type (as defined in Section), to be a path in the graph that might only take the form: $v_1 \xrightarrow{R_1} v_2 \xrightarrow{R_2} \ldots \xrightarrow{R_n} v_n$, for $v_i \in V$. Given a weighted heterogeneous network $G(V, E, L)$, and a metapath scheme p, the transition probability at step i is defined as follows:

$$P(v_{i+1} | v_i, p) = \begin{cases} \frac{weight(v_{i+1}, v_i)}{\Sigma_{v \in NbL(v_i, R_i)} weight(v^i, v)}, & \text{if } l(v_{i+1}, v_i) = R_i \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where $NbL(u, v, R)$ is the set of nodes that are connected to u via an edge of label R.

A similar extension to node2vec, creating metapath-based random walks, was suggested by metapath2vec (Dong, Chawla, and Swami 2017) for unweighted and unlabeled graphs. However, this approach requires defining a single metapath, which is based on some prior knowledge of the problem domain. For our problem, no such definition has been previously studied. We next discuss how to create several metapaths in an unsupervised manner. The algorithm greedily selects the metapaths best suited per each novel molecule. Finally, an embedding is created specifically for the novel molecule from the different metapath-based random walks.

Metapath Selection We begin by randomly generating metapaths according to the node and edge types in the heterogeneous network. Then, each potential metapath creates a candidate embedding for each node (Figure 1). To create the final embedding of each node in G, we present a greedy algorithm described in the next section (Section Embedding Algorithm): given a novel molecule, mol, select the "best" set of metapaths from which the final embeddings of all the graph nodes can be created. Once metapaths are selected, the final embedding of the nodes is the average of the metapath-corresponding embeddings. In the next section, we will further describe how to define the notion of "best set" and select the metapaths from which the final embedding of each node is created.

Embedding Algorithm We use beam search for creating the set of metapaths from which the final embedding for each node will be created. The algorithm holds a set of the best metapaths identified so far, $mset$. At each step, it selects an additional metapath that maximizes a given objective function. Our objection function is dependant on the novel molecule given as a query, which we present next.

Intuitively, we assume that if we knew the text representation of the novel molecule, we could leverage it for document ranking. That is, if we knew the name of the molecule when it will be discovered in the future, we could use text-ranking techniques to identify relevant documents. We therefore devise a function estimating how similar the distances between the node's embeddings in the graph and the novel molecule are compared to the distances of the text embeddings of those nodes and the estimated text representation of the molecule. Formally, let t be the desired future text representation of the novel molecule mol (i.e., the name of the molecule when it will be discovered in the future). Since we aim at ranking text documents, we wish to identify the known molecules, whose embeddings are closest to t.

Let $list_{text}$ be a sorted ranked list of molecules by their similarity to t, and $list_{graph}$ be a sorted ranked list of molecules by their similarity to the the graph embedding set of mol, as given by the metapath set $mset$. We define the objective function, $f_{mol,mset}$, as:

$$f_{mol,ms} = \Sigma_{i=1}^{n}\Big(sim(t, list_{text}[i])) - sim(t, list_{graph}[i])\Big)/i$$

where k is a parameter of the model. Intuitively, we evaluate the similarity by the text representation and wish the graph embedding to be as close as possible to that similarity. We chose cosine similarity as the similarity function, sim. We select an approximation of t by a fully connected neural network that receives as input the mol2vec representation of mol and predicts the word2vec representation of the known molecule. The details of the architecture are given by (Kim 2014). The classifier is trained using molecules from the training set, for which both mol2vec and word2vec representation exist. We refer to the application of this method to the molecule-document graph as *Selective molgraph2vec*.

Molecular Ranking Algorithms

In this section, we explore several algorithms that leverage the embeddings described in the previous sections for the purpose of ranking documents for novel molecule.

Bag of Molecular Entities Ranking (BOME). For each molecule whose name is mentioned in the document d we measure the distance of the embeddings, and calculate the final ranking score by: $score(d, mol) = \sum_{m \in d} sim(e(m), e(mol))/|d|$, where $|d|$ is the number of molecules mentioned in d, sim is a similarity function, and $e(mol)$ is the embedding of mol. Cosine similarity is used for sim in our experiments.

Pointwise Document-Molecule Ranking. We adopt a pointwise approach to rank molecule-document pairs. The score of a document d and a novel molecule mol is defined as the score the classifier assigns given their embeddings, here $e(d)$ is the embedding of the document d:$score(d, mol) = classifier(e(d), e(mol))$. Specifically in this work, we follow the practice of (Severyn and Moschitti 2015) and use a convolutional neural network (CNN) with mean square error as the loss function.

Molecular KNN Ranking. We define $closest(mol_{novel}, k)$ to be the set of k most similar molecules to novel molecule mol_{novel} from the set of known molecules. To define similarity to identify the closest molecules, we use the textual representation of the molecules in the set $closest(mol_{novel}, k)$, and use the cosine similarity of their embeddings (as defined in Section) to rank the medical papers. Additionally, we define the similarity of known molecules and papers to be cosine similarity of their word2vec representations. The similarity of mol_{novel} and a paper, p, depends on the similarity between the novel molecule and k known intermediate molecules, $mol_{intermediate} \in closest(mol_{novel}, k)$ and the similarity of those k known intermediate molecules and the medical document, d:

$$score(M, d) = \sum_{m \in \text{closest}(M,k)} \text{cosSim}(\text{e}(M), \text{e}(m)) \cdot \text{cosSim}(\text{W2V}(m), \text{W2V}(d))$$

where M is a novel molecule, $cosSim$ is the cosine similarity function, e is a molecule embedding method (Section), and k is a hyperparameter.

Molecular KNN++ Ranking. We extend the Molecular KNN algorithm to consider additional similar molecules and refer to the new algorithm as Molecular KNN++. In this version, we define $closest(mol_{known}, l)$ to be a function of a known molecule mol_{known} that returns the set of l most similar molecules to mol_{known} from the set of known molecules. At this stage, since we are dealing only with known molecules that have a textual context, we define the similarity between them to be the cosine similarity between their word2cec representation. Intuitively, we want to leverage the similarity between word2vec representation of known molecules to create a less biased ranking model:

$$\text{score}(M, d) = \sum_{m \in \text{closest}(M,k)} \text{cosSim}(\text{e}(M), \text{e}(m)) \cdot \sum_{x \in \text{closest}(m,l)} \text{cosSim}(\text{W2V}(x), \text{W2V}(d))$$

where M is a novel molecule, d is a medical document, $cosSim$ is the cosine similarity function, e is a molecule embedding method (Section), and k and l are hyperparameters.

Experimental Setting

PubMed is a bibliographic database dealing with life, medical, and paramedical sciences (Falagas et al. 2008). The database contains bibliographic records of most scientific articles published since the 1950s (and sometimes earlier) in all languages. The PubMed dataset includes 29,700,000 papers. For our experiments, we consider only papers with a full abstract, reducing the set to 11,954,865 papers. We limit the papers in our dataset to those published in journals with impact factor of 5 or higher, reducing the set to 1,429,705 papers. For queries, we use a list of well-known drugs that includes 1286 drug names with a notation of their chemical structure (using SMILES). Our experimental setup simulates a situation wherein the current time is the end of the year 2000. We therefore only have access during training to PubMed papers published until the end of 2000. We consider the invention date of a drug as the publication date of the first PubMed paper that refers to it. We consider all drugs invented until the end of the year 2000 for the training set (1105 drugs) and the rest as the test set used as novel molecules for queries (181 drugs). For each drug in the training set, we create a word2vec representation using a skip-gram model based on all the PubMed abstracts that refer to it until year 2000.

Gold-Standard Test Set: Intuitively, in the test set we are given a SMILES of a novel never-before-seen molecule and wish to evaluate whether the document ranking for it is relevant. In our experiments, we use drugs not known in the year 2000 and compare to the ranking of the molecules once invented and named after the year 2000. In other words, we compare the ranking of the documents for the SMILES until the year 2000 to the future document ranking for the name of the molecule after the year 2000. The gold-standard ranking of the PubMed papers for a given drug, we consider the cosine similarity between the word2vec representation of the name of the molecule and the document representation as suggested by Arora, Liang, and Ma (Arora, Liang, and Ma 2017).

For evaluation, we use three ranking metrics: MAP, precision, and NDCG (Manning, Raghavan, and Schütze 2010). We inspect precision and NDCG at the top 10, 50, and 100 results. For the MAP and precision metrics, we consider the top 100 gold standard ranking results for each drug in the test set as the relevant results, while the other papers are considered as irrelevant. For NDCG, we consider the cosine similarity score between the molecule and the document.

In addition to the methods presented in Section , we present results of a random baseline which, given a molecule and a paper, assigns an random number between 0 and 1 as their similarity and produces the ranking accordingly.

For each embedding method and for each ranking algorithm, we learn on a validation set derived from the training set with a grid search the best value of all the hyperparameters, including embedding size, the number of metapaths, and length of the paths.

Experimental Results

Table 1 presents standard information retrieval evaluation metrics (Sanderson 2010): MAP, precision, and NDCG across the four algorithms, each with the four embedding methods. Best results, marked in bold are achieved via the mKNN++ algorithm and Selective molgraph2vec embedding. Selective molgraph2vec embedding shows superior results for each algorithm tested. When comparing the different ranking algorithms we observe that the BOME method performs, as expected, better than the random baseline, across all metrics. The supervised Pointwise approach achieves better results than BOME. The mKNN approach achieves a substantial performance gain over the Pointwise approach. Across most ranking methods, embeddings created with metapath selection yield better results.

Exploring mKNN++

We hypothesize that our algorithm preforms better on queries of drugs for which there is sufficient information in the training set, i.e., drugs with similar molecular structures to the query molecules exist in the graph and there is a large number of documents for these existing drugs.

We divide our test set into ten groups according to the Precision@10 score. Group 0 has the lowers score of 0 precision@10, group 1 has 0.1 precision, and so on. Due to the limited size of the test set and the diversity of drugs, we cannot offer a thorough statistical analysis of the differences between groups. We will however, discuss some interesting traits relaxing claims of statistical power. The entire test set consists of 181 drugs of diverse types. We focus on drug categories that can be identified in the test set and contain a minimal number of 7 drugs: Chemotherapy/Cancer, Antiviral/HIV, Antibiotic, Diabetes, and Anti-Hypertensive. Table 2 presents the percentage of each drug category in the precision groups. We assume these results mirror the research trends in our training data. If a category is a popular research subject, the dataset will contain more examples of papers discussing it allowing for better discovery.

Algorithm	Embedding	MAP	Precision @10	Precision @50	Precision @100	NDCG @10	NDCG @50	NDCG @100
Random	–	0.0052	0.0055	0.0062	0.0050	0.5997	0.6274	0.6435
BOME	mol2vec	0.0241	0.0795	0.0477	0.0379	0.7246	0.7248	0.7304
	node2vec	0.0245	0.0784	0.0542	0.0406	0.7126	0.7113	0.7106
	molgraph2vec	0.0290	0.0872	0.0596	0.0432	0.7164	0.7209	0.7237
	Selective molgraph2vec	0.0340	0.0872	0.0710	0.0517	0.7437	0.7479	0.7476
Pointwise	mol2vec	0.0651	0.1046	0.0888	0.1359	0.7779	0.7983	0.8109
	node2vec	0.0688	0.1408	0.1097	0.1380	0.7800	0.8035	0.8163
	molgraph2vec	0.0671	0.1500	0.1123	0.1432	0.7850	0.8075	0.8243
	Selective molgraph2vec	0.0736	0.1600	0.1252	0.1471	0.7970	0.8155	0.8321
mKNN	mol2vec	0.2287	0.3226	0.2771	0.2188	0.8092	0.8258	0.8334
	node2vec	0.2211	0.3071	0.2701	0.2133	0.8040	0.8210	0.8284
	molgraph2vec	0.2139	0.3475	0.2797	0.2229	0.8404	0.8550	0.8620
	Selective molgraph2vec	0.2379	0.3883	0.3072	0.2428	0.8556	0.8674	0.8743
mKNN++	mol2vec	0.1982	0.3292	0.2647	0.2080	0.8431	0.8589	0.8660
	node2vec	0.2156	0.3558	0.2843	0.2233	0.8468	0.8610	0.8675
	molgraph2vec	0.2335	0.3712	0.3087	0.2420	0.8532	0.8676	0.8740
	Selective molgraph2vec	**0.2465**	**0.3928**	**0.3237**	**0.2511**	**0.8616**	**0.8742**	**0.8797**

Table 1: MAP, precision, and NDCG for article ranking per molecular query across different algorithms and embedding methods.

Group	Size	Chemo	Antiviral	Antibiotic	Other	Diabetes	Hypertension
All data	181	18%	6%	7%	61%	4%	3%
0	52	30.7%	1.9%	5.7%	55.7%	5.7%	0
1	24	41.6%	4%	0	5%	4%	0
2	9	0	11%	0	77.8%	11%	0
3	12	8.3%	0%	0	91.7%	0	0
4	6	16.7%	0	0	8.3%	0	0
5	9	0	0	0	88.8%	11%	0
6	11	9%	0	0	72.3%	0	18%
7	9	11%	11%	11%	44%	11%	11%
8	15	13%	0	13%	60%	6.7%	6.7%
9	34	2.9%	20.6%	17.6%	52.9%	0	5.9%
0+1 (low)	85	31%	4%	4%	56%	6%	0
8+9 (high)	49	6%	14%	16%	55%	2%	6%

Table 2: Precision Groups

Qualitative Example

Clofarabine is a chemotherapy drug approved by the FDA in 2004. It is used for treatment of Acute Lymphoblastic Leukemia (ALL) in pediatric patients. We mimic a research process that could have been conducted in 2000 using our system. The molecular representation of Clofarabine was fed to the ranking algorithm, which ranked the following titles relating to this molecule: 'Effects of mitoxantrone in combination with other anticancer agents on a human leukemia cell line.', 'In vitro cytotoxic effects of fludarabine (2-F-ara-A) in combination with commonly used antileukemic agents by isobologram analysis.','Myelodysplastic syndrome following successful therapy of acute promyelocytic leukemia.'. These all refer to initial experiments in different types of cancers. The similarity was observed due to the similarity of the molecular structure of Clofarabine to drugs such as Mitoxantrone, which was discovered to treat certain types of cancer.

Conclusions

In this work, we suggest a solution to the task of ranking documents based on a novel molecule, given as a query. Motivated by the idea of combining the structure of the molecule and its textual context, our solution creates embeddings for drugs based on their molecular structure, and embedding for documents based on their text. Which construct one heterogeneous graph. We traverse this graph, generating an embedding for a novel molecular structure, and retrieving relevant documents based on this new embedding. We deployed our system in a material discovery setting targeted at searching for known materials with unexpected/unexplored properties. It is currently deployed at a pharmaceutics research lab focused on personalized medicine and targeted drug delivery. Future work on this subject may include further investigation of heterogeneous graph-based embedding for molecules, for instance by using graph representations of molecular structures instead of the SMILES-based mol2vec.

References

Arora, S.; Liang, Y.; and Ma, T. 2017. A simple but tough-to-beat baseline for sentence embeddings. *ICLR'17*.

Bhagat, S.; Cormode, G.; and Muthukrishnan, S. 2011. Node classification in social networks. *Social network data analytics*, 115–148.

Cao, S.; Lu, W.; and Xu, Q. 2015. Grarep: Learning graph representations with global structural information. *CIKM'15*.

Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; and Jensen, K. F. 2017. Convolutional embedding of attributed molecular graphs for physical property prediction. *JCISD8'17*, 57(8): 1757–1772.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. *SIGKDD'17*.

Falagas, M. E.; Pitsouni, E. I.; Malietzis, G. A.; and Pappas, G. 2008. Comparison of PubMed, Scopus, web of science, and Google scholar: strengths and weaknesses. *FASEB'08*, 22(2): 338–342.

Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. O.; and Baker, N. 2017. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. *arXiv preprint arXiv:1706.06689*.

Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; and Aspuru-Guzik, A. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS'18*, 4(2): 268–276.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. *SIGKDD'16*.

Harel, S.; and Radinsky, K. 2018. Prototype-based compound discovery using deep generative models. *Molecular pharmaceutics*, 15(10): 4406–4416.

Jaeger, S.; Fulle, S.; and Turk, S. 2018. Mol2vec: Unsupervised machine learning approach with chemical intuition. *JCISD8'18*, 58(1): 27–35.

Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; and Zhavoronkov, A. 2017. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics*, 14(9): 3098–3104.

Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; and Riley, P. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8): 595–608.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. *EMNLP'14*.

Liben-Nowell, D.; and Kleinberg, J. 2007. The link-prediction problem for social networks. *JASIST'07*, 58(7): 1019–1031.

Manning, C.; Raghavan, P.; and Schütze, H. 2010. Introduction to information retrieval. *NLE'10*, 16(1): 100–103.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *NIPS'13*.

Olivecrona, M.; Blaschke, T.; Engkvist, O.; and Chen, H. 2017. Molecular de-novo design through deep reinforcement learning. *JCOHB3'17*, 9(1): 48.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. *SIGKDD'14*.

Rogers, D.; and Hahn, M. 2010. Extended-connectivity fingerprints. *JCISD8'10*, 50(5): 742–754.

Sanderson, M. 2010. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi+ 482 pages. *Natural Language Engineering*, 16(1): 100–103.

Schneider, N.; Fechner, N.; Landrum, G. A.; and Stiefl, N. 2017. Chemical Topic Modeling: Exploring Molecular Data Sets Using a Common Text-Mining Approach. *JCISD8'17*, 57(8): 1816–1831.

Severyn, A.; and Moschitti, A. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *SIGIR'15*.

Spangler, S.; Wilkins, A. D.; Bachman, B. J.; Nagarajan, M.; Dayaram, T.; Haas, P.; Regenbogen, S.; Pickering, C. R.; Comer, A.; Myers, J. N.; 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. ACM.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB'11*, 4(11): 992–1003.

Suresh, P.; and Basu, P. K. 2008. Improving pharmaceutical product development and manufacturing: impact on cost of drug development and cost of goods sold of pharmaceuticals. *Journal of Pharmaceutical Innovation*, 3(3): 175–187.

Swanson, D.; and Smalheiser, N. 1999. Implicit Text Linkages between Medline Records: Using Arrowsmith as an Aid to Scientific Discovery. *Library Trends*, 48(1): 48–61.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. *WWW'15*.

Wan, F.; and Zeng, J. 2016. Deep learning with feature embedding for compound-protein interaction prediction. *bioRxiv'16*.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.