Using Knowledge Graph Embeddings from Biomedical Language Models to Infer Drug Repurposing Candidates for Rare Diseases

Stanford CS224N Custom Project

Yash Patil
SUNet ID: ypatil
Department of Computer Science
Stanford University
`yashpatil@stanford.edu`

John Wang
SUNet ID: jwang003
Department of Computer Science
Stanford University
`jwang003@stanford.edu`

Abstract

Approaches for finding new drugs are often costly and ineffective. On the other hand, available drugs that are currently being used to treat certain diseases could potentially be repurposed for others. Recent studies have attempted to represent known drug-disease interactions as networks in order to predict unknown edges that might correspond to novel drug-disease interactions (Kim et al., 2022). In particular, one method to do this is to create an embedding representation of a knowledge graph containing known drug-disease interactions and perform link prediction to predict new edges in the graph. Recently, several embedding methods have leveraged the use of language models to take advantage of both the structure of knowledge graph and its textual components when creating representations (Yao et al., 2019) At the same time, open source NLP tools have enabled the creation of large networks of biomedical relationships (Percha and Altman, 2018). In this project, we modify existing state of the art (SOTA) language model embedding methods to biomedical applications and frame the drug repurposing problem as a knowledge graph completion task. We introduce three new findings: (1) a novel dataset of known drug-disease treatments derived from the Global Network of Biomedical Relationships (GNBR) to test network based prediction tasks for drug repurposing, and (2) a knowledge graph embedding method that leverages language models and the existing biomedical corpora (Percha and Altman, 2018), achieving improved performance over SOTA baselines, and (3) a faster variant of BioBERT Lee et al. (2020) called biobert-tiny, pretrained on six million PubMed abstracts and publicly available on Hugging Face[1]. The code for this project is open source[2].

1 Key Information

- Mentor: Elaine Sui
- External Collaborators (if you have any): Russ B. Altman
- Sharing project: No

2 Introduction

The goal of drug discovery is to synthesize new compounds to treat diseases. However, the process of creating and testing novel drugs is slow and expensive. Rare diseases compound this problem with

[1] `https://huggingface.co/yashpatil/biobert-tiny-model`

[2] `https://github.com/jwang307/gnbr-project`

low patient volume, generally less knowledge on treatments, and a lack of incentives for companies to research these diseases. Even though individual rare diseases affect less than 0.05% of the population by definition, over 100 million people worldwide are affected[3]. Drug repurposing, the application of clinically approved drugs on different diseases than originall intended, offers a less expensive process to finding cures when compared with drug discovery. Databases of drug-disease interactions such as GNBR have enabled drug repurposing methods that leverage knowledge graphs - networks consisting of entities represented by nodes and relationships between entities represented by edges. Knowledge graphs are often represented by triples of the form (h, r, t), where h and t are two entities in the graph known as the head and tail, and r is the relation between them. The following is an example of a triple extracted from GNBR:

$$(h, r, t) = (\texttt{JNJ-39758979}, \texttt{alleviates}, \texttt{asthma})$$

In link prediction, the goal is to find valid triples (h, r, t) that do not exist in the graph. In this project, we model the drug repurposing problem as a link prediction task for finding triples of the form $(drug, \text{treatment}, disease)$. Specifically, we run a subvariant of link prediction called triple classification, where triples (h, r, t) in the test set have a corresponding label 0 or 1, and the model tries to classify each triple as valid or invalid. Popular methods for this include TransE by Bordes et al. (2013), TransH by Wang et al. (2014), and DistMult by Yang et al. (2015), which learn low-dimensional vector representations for entities and relations in the graph. Once the embeddings are generated, similarity metrics like cosine similarity, dot product, or Euclidean distance are used to measure the strength of relationships between entities, which in turn helps in predicting the missing links.

Recently, large databases of biomedical relationships such as PharmKG and GNBR have introduced knowledge graphs on the order of millions of interactions between drugs, genes, and diseases (Zheng et al., 2020)(Percha and Altman, 2018). This prompts the question as to whether language models can leverage this increase in available data to more accurately predict drug-disease interactions. While using language models to produce knowledge graph embeddings is not a novel concept, it has never been applied to drug repurposing. In this project, we use a knowledge graph embedding method that leverages language models for our drug repurposing task, achieving improved performance over standard baseline methods on our dataset. The main contributions of this project are as follows:

- We show a **2.63%** improvement in accuracy over baseline methods using BioBERT Lee et al. (2020) to generate knowledge graph embeddings for triple classification.
- We introduce a new dataset for rare disease interactions with known drugs and genes extracted from GNBR and consisting of 25411 unique entities, 31 types of relations, and 166859 edges.
- We introduce a new language model called biobert-tiny, which was pretrained on six million PubMed abstracts and runs over eight times faster than BioBert.

3 Related Work

This project consisted of work in knowledge graph embeddings, biomedical language model pretraining, and biomedical dataset creation. As such, related work in each area is described below.

3.1 Knowledge Graph Embeddings

A core aspect of the project involves the conversion of triples into knowledge graph embeddings. Formally, this task can be modeled as an operation from $(h, r, t) \rightarrow \mathbf{h} \in \mathbb{R}^d, \mathbf{r} \in \mathbb{R}^d, \mathbf{t} \in \mathbb{R}^d$, where d is the embedding dimension. There are two main types of methods for embedding triples into vector spaces: structure-based and description-based knowledge embeddings (Wang et al., 2022).

In structure-based embedding methods, calculations are based solely on the nodes and relation of the triple. Among structure-based methods, a landmark technique called TransE uses a geometric approach to scoring triples (Bordes et al., 2013). TransE frames relations as translations in the embedding space, scoring a triple by the distance of the vector for $\mathbf{h}$ from the vector for $\mathbf{t}$ after a translation of $\mathbf{h}$ defined by $\mathbf{r}$. Further modifications of TransE include TransH and TransR, which

[3] `https://www.orpha.net/`

propose differences in the translation operation (Wang et al., 2014)(**?**). In TransH, both $\mathbf{h}$ and $\mathbf{t}$ are projected to a relation-specific hyperplane, while in TransR, $\mathbf{h}$ and $\mathbf{t}$ exist in entity-specific spaces and are projected to a relation-specific vector space.

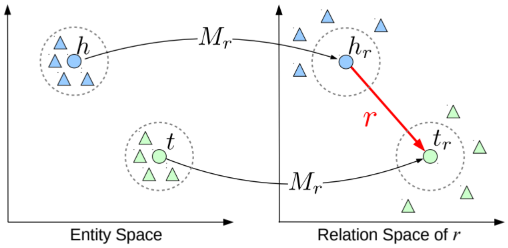


Figure 1: Simple Visualization of Entity and Relation Spaces in TransR

The other prominent type of structure-based methods are known as semantic matching methods or tensor decomposition models. These methods treat the knowledge graph as an adjacency matrix represented by a three dimensional tensor, which can be decomposed with the embeddings for entities and relations. In knowledge graph completion, these methods predict new links as values that are close to 1 in the adjacency matrix. A popular method for this is DistMult, which treats $\mathbf{r}$ as a $d \times d$ diagonal matrix, and the resulting score is the matrix product of **h, r** and $\mathbf{t}$.

Recently, description-based embedding methods have become popular because they leverage language models to better embed entities and relations. While structure-based methods rely solely on the existence of triples, description-based methods are able to use textual knowledge of entities via pretrained language models to generate embeddings. In addition, most description-based methods also introduce descriptions for each entity and relation. In theory, these descriptions for entities and relations enable a better representation of the triple and a more accurate embedding. A popular description-based method is KEPLER by Wang et al. (2020), a language model designed and trained on both knowledge graph completion and general language model tasks. Yao et al. (2019) published the landmark description-based method KG-BERT, which uses BERT to generate a sequence embedding for the triple and scores the sequence embedding with a sigmoid classifier. Most recently, Wang et al. (2022) introduced LMKE, removing the requirement of descriptions for each entity and relation by concatenating the triple and removing [SEP] tokens between entities. In practice, this increases the accessibility of these methods, as real-world datasets like the one created in this project do not have readily available descriptions for entities and relations. To the best of our knowledge, LMKE produces SOTA results on general knowledge graph completion benchmarks.

3.2 Biomedical Language Models

Biomedical language models have gained significant attention in recent years, as they enable more accurate representation of domain-specific concepts and relationships. These models are typically pre-trained on large-scale biomedical text corpora, such as PubMed abstracts or full-text articles, and then fine-tuned for specific tasks. Notable models include *BioBERT* (Lee et al., 2020), *SciBERT* (Beltagy et al., 2019), and *BlueBERT* [4]. These models are pre-trained on large-scale biomedical text corpora, such as PubMed abstracts or full-text articles, and fine-tuned for specific tasks.

BioBERT and *SciBERT* pioneered the adaptation of the BERT architecture to the general biomedical domain, demonstrating substantial improvements in various biomedical NLP tasks, such as named entity recognition, relation extraction, and question-answering (Lee et al., 2020). More specific models like *BlueBERT* which focuses on molecular mechanisms of diseases and therapeutics development can be used for better results on certain tasks.

3.3 Biomedical Datasets

There are many online databases cataloguing known drug, disease, and gene interactions. Traditional databases such as the Comparative Taxicogenomics Database (CTD) have been manually curating

[4] `https://github.com/ncbi-nlp/bluebert`

interactions since 2004 and contain over 40 million interactions between drugs, genes, and diseases along with other areas of interest such as exposure events. A useful database for rare diseases is Orphanet[5], which manually catalogues interactions between known drugs and rare diseases. OMIM[6] is an online compendium of genes and genetic disorders created by the School of Medicine at Johns Hopkins University, and MeSH is a vocabular database monitored by the National Library of Medicine (NLM) at the National Center for Biotechnology Information (NCBI).

Since the onset of NLP tools such as CoreNLP by Manning et al. (2014) and biomedical language models such as BioBERT, multiple knowledge graphs have been created that leverage NLP to mine massive text copora. Notable ones include GNBR by Percha and Altman (2018), which mined over 16 million PubMed abstracts with a variety of NLP tools including PubTator for named entity recognition, CoreNLP for dependency path parsing, and a clustering algorithm by Percha and Altman (2015) to create a final knowledge graph of drug, gene, and disease interactions characterized by 36 types of relations and over 2 million triples. More recently, Chandak et al. (2023) used multiple manually curated databases such as DrugBank and the Disease Gene Network (DisGeNet) to build PrimeKG, knowledge graph of over 17,000 diseases associated with various biological entities and events. Even so, there is a lack of available benchmarks for biomedical knowledge graph completion. The most accessible is PharmKG by Zheng et al. (2020), a benchmark for biomedical knowledge graph completion and mining with over 500 thousand edges and eight thousand entities across 29 relation types.

4 Approach

4.1 Dataset Generation

Because our task focused on drug-disease treatment interactions for rare diseases, we found triples in PharmKG and other knowledge graph benchmarks too sparse for meaningful testing (< 1000 drug-disease treatment pairs for rare diseases). We introduce a new dataset for triple classification of rare disease interactions, extracted from data in GNBR. To do this, we first processed GNBR to a list of triples (h, r, t) and mapped each relation to their textual definition. Following this, we compiled a list of MeSH IDs corresponding to rare diseases as defined by Orphanet. Using the list of diseases from CTD, we matched Orphanet IDs to CTD diseases to obtain a list of clean disease names. In GNBR, we created mapped diseases to MeSH IDs and cleaned the formatting of diseases since multiple diseases were found with the same MeSH ID. For each rare disease extracted from CTD, we mapped that disease to the closest matching entity in GNBR using the Ratcliff-Obershelp algorithm with a cutoff of 0.3. This was necessary because of inconsistent naming in GNBR (Ex. `acute_myeloid_leukemia` vs `myeloid leukemia, acute`). We obtained 1279 rare diseases from GNBR through this cross-matching method.

Using the list of rare diseases, we created our dataset using breadth first search and a depth of 2. In order to reduce our dataset to a computationally feasible size, we capped the degree of each node to 120 and randomly sampled if the degree of that entity in GNBR was greater. The size of our final dataset is shown below in comparison to PharmKG.

Dataset	Edges	Entities	Relations	Average Degree
GNBR	166859	25411	31	3.28
PharmKG	500958	7603	29	32.94

Table 1: Comparison of our dataset and PharmKG

Since our objective is focused on rare diseases, the "rarity" is reflected in the sparsity and low degree average of the graph.

[5] https://www.orpha.net/consor/cgi-bin/index.php

[6] https://omim.org/

4.2 biobert-tiny

A key inspiration for our work was BioBERT Lee et al. (2020), a BERT-based language model that was pre-trained on large-scale biomedical corpora. In conducting our research, we were constrained on both compute power and processing time. Therefore, we focused on creating a smaller, more efficient model tailored to the task of inferring novel drug candidates for diseases, which we call biobert-tiny. This smaller model allows for faster training and inference times, while still leveraging the domain-specific knowledge from the large-scale biomedical text corpora. To develop a smaller and faster variant of BioBERT, we performed the following steps:

1. **Data Collection:** We extracted approximately 6000000 paper abstracts from the online MEDLINE/PubMed Baseline Repository (MBR). Abstracts were chosen at random from the past 10 years of data.
2. **Tokenizer Training:** In order to achieve optimal accuracy, we created a custom tokenizer. We initialized the tokenizer with the pre-trained "bert-base-uncased" tokenizer to reuse its special tokens. Then, we trained the tokenizer with Hugging Face's default byte pair encoding (BPE) algorithm on the raw dataset with a vocabulary size of 32,000 to account for new tokens found in the biomedical corpus. This trained tokenizer can be found on Hugging Face.[7]
3. **Data Preprocessing:** Using our custom tokenizer, we tokenized the input and applied a simple truncation strategy where documents longer than 512 tokens were truncated without splitting them into several documents. The preprocessed dataset was then shuffled and pushed to the Hugging Face Hub for later use.
4. **Model Pre-training:** To pre-train the biobert-tiny model, we forked a smaller version of BERT called BERT-tiny[8] (4.4M parameters) and employed Masked Language Modeling (MLM), a technique by Devlin et al. (2019) that encourages bidirectional learning from text. In MLM, a word in a sentence is masked (hidden), and the model is tasked with predicting the masked word using the context provided by the surrounding words. See the example below:

```
Mitochondria are tiny [MASK] inside cells that are involved in
releasing energy from food.
```

4.3 LMKE and Triple Classification

To perform triple classification, we used LMKE[9] by Wang et al. (2022) with modifications. We chose LMKE for its SOTA results on general knowledge graph completion and its ability to generate embeddings without required textual descriptions for each entity and relation. In this section, we formally provide an overview of LMKE's architecture as well as our modifications.

LMKE: LMKE takes in a triple $u = (h, r, t)$ and concatenates it to form the input to the tokenizer, [CLS], h, r, t, [SEP]. If descriptions - denoted d_h, d_r, d_t - are included, then the input to the tokenizer is [CLS], h, d_h, r, d_r, t, d_t, [SEP]. This sequence is passed through the tokenizer of the chosen language model, which is passed into the language model. We tested three language models with decreasing size:

- BioBert: 110M parameters: $H = 12$, $d = 768$, 110M parameters
- TinyBioBert[10]: $H = 4$, $d = 768$, 15M parameters
- biobert-tiny: $H = 2$, $d = 128$, 4.4M parameters

The architecture and training of the first two models is described in Devlin et al. (2019), while our pretrained model is described in this paper. To aggregate the output embeddings from the language

[7] https://huggingface.co/yashpatil/biobert-tiny-tokenizer

[8] https://huggingface.co/prajjwal1/bert-tiny

[9] https://github.com/Neph0s/LMKE

[10] https://huggingface.co/nlpie/tiny-biobert

model, we use the embedding of [CLS], denoted $\mathbf{c} \in \mathbb{R}^{\mathbf{d}}$. This embedding serves as the embedding vector of the triple u. To score the triple, we use a single layer classifier:

$$\phi(u) = \sigma(\mathbf{wc} + \mathbf{b})$$

Here, $\mathbf{w} \in \mathbb{R}^{\mathbf{d}}$ and b are learnable parameters.

Support Scores: In the original LMKE paper, the labels for a triple are either 1 for true or 0 for negative triples. We modify the labels used for training LMKE by leveraging support scores found with each triple in GNBR. Because GNBR was mined with software tools and not manually curated, each triple in the network comes with a support score s, normalized between 0 and 1, to indicate the uncertainty of the triple being true, where scores closer to 1 are more certain as fact. These scores can be leveraged using uncertainty knowledge embedding methods such as Chen et al. (2019), but we introduce a simple modification to LMKE by modifying the label of each training triple to s, representing the probability that the triple is true. During testing and vaildation, we ensure that triples tested are all high confidence (> 0.95) and return to predicting 1 or 0 for true or false triples.

Descriptions: Because many entities in GNBR are rarely studied drugs, genes, or diseases, scraping descriptions off the online PubMed corpus is difficult and time-consuming. A main advantage of LMKE is it's ability to work without descriptions; even so, we test the ability of the model on different configurations of input. We define d_h, d_r and d_t as the "clean" strings of the entities or relations. For example, if $h =$ `3-methylcrotonyl_Coa_carboxylase_1_deficiency`, we define $d_h =$ `3-methylcrotonyl CoA carboxylase 1 deficiency`. We then configure the input of the triples in three ways to see if simple textual descriptions improve performance:

- [CLS]$, h, r, t,$ [SEP]
- [CLS]$, d_h, d_r, d_t,$ [SEP]
- [CLS]$, h, d_h, r, d_r, t, d_t,$ [SEP]

Baselines: We ran TransE, a geometric method, and DistMult, a tensor decomposition method, as baselines for triple classification (Bordes et al., 2013)(Yang et al., 2015). In addition to their respective publications, these methods are briefly described above.

We ran DistMult with the following hyperparameters:

```
Embedding size:200 | Batch size:1658 | learning rate:0.5 | Epochs:1000
```

We ran TransE with the following hyperparameters:

```
Embedding size:200 | Batch size:1658 | learning rate:1.0 | Epochs:1000
```

In order to test the effectiveness of biomedical language models compared to general language models, we also ran LMKE with BERT-tiny as a baseline.

5 Experiments

5.1 Data

The dataset generation process is described above. To split the dataset into train, validation, and test sets, we used a $10\%, 20\%, 70\%$ split among all drug-disease treatment edges. This is because our objective was to identify triples of the type (drug, treatment, disease) In addition, we require all entities in the validation and test sets to be in the training set so we don't intorduce new nodes during testing. To do this, we first extracted all (drug, treatment, disease) triples. Because there were only 1085 triples of this type, the majority of the split was dedicated to testing, resulting in 217 positive triples for the validation set and 760 positive triples for the test set. All other triples were included in the train set, resulting in 165882 positive triples in the training set. We then confirmed that all entities in the validation and test set were present in the training set. Finally, we generated negative triples with a negative sampling rate of 1, randomly corrupting the head or tail of every triple to an unseen triple. The final splits total 331764 triples in the training set, 434 triples in the validation set, and 1520 triples in the test set.

5.2 Evaluation method

Since triple classification is a binary classification task, we use standard metrics, namely accuracy, precision, recall, and F1 score. We report accuracy since the dataset has a negative sampling rate of 1, resulting in a balanced test set. Given True Positive $= T_p$, True Negative $= T_n$, False Positive $= F_p$, False Negative $= F_n$:

$$accuracy = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{1}$$

$$precision = \frac{T_p}{T_p + F_p} \tag{2}$$

$$recall = \frac{T_p}{T_p + F_n} \tag{3}$$

$$F_1 = 2 * \frac{precision\ *\ recall}{precision\ +\ recall} \tag{4}$$

5.3 Experimental details

We ran LMKE with several tuned hyperparameters:

- Batch Size: $\{16, 32, 64\}$
- Descriptions: {triples, triples + descriptions, descriptions}
- Support Scoring: {`True, False`}

We ran all experiments for 50 epochs with 0.0005 learning rate, 10^{-7} weight decay, and a negative sampling rate of 1. We trained each model with Binary Cross Entropy Loss. If s is the score of a triple, y is the true label, D^+ is the training set, and D^- is the negative sampled training set, our loss function is given by

$$\mathcal{L} = - \sum_{i \in \{D^+ \cup D^-\}} (y_i \log(s_i) + (1 - y_i) \log(1 - s_i)) \tag{5}$$

At the end of each epoch, parameters were saved if the validation accuracy outperformed all previous epochs. The best performing parameters were used for testing. We selected the hyperparameters for each language model by choosing the configuration with the best validation set accuracy. The best performing hyperparameters are shown with each model.

LMKE with BioBERT:

```
Batch Size:64 | Descriptions:triples+descriptions | Support Scoring:False
```

LMKE with TinyBioBERT and biobert-tiny:

```
Batch Size:64 | Descriptions:triples+descriptions |Support Scoring:False
```

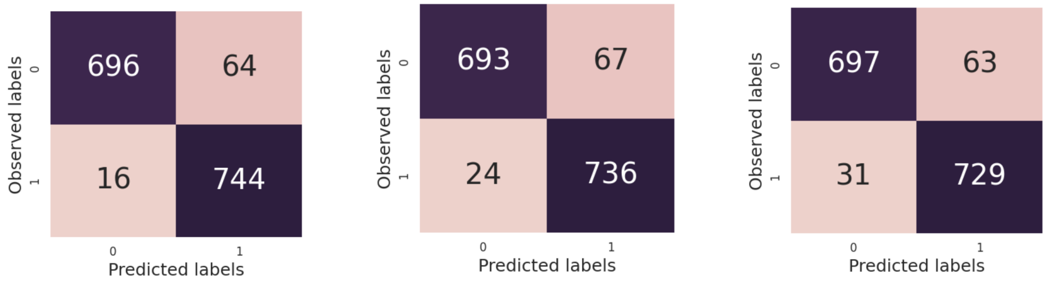
Experiments were performed on an AWS EC2 instance with one NVIDIA A100 GPU and on Sherlock with slurm jobs using 4 NVIDIA A40 GPUs.

5.4 Results

In all metrics, LMKE with BioBERT as the language model was the highest performing method. Furthermore, all BioBERT variants outperformed the baselines in accuracy and F1 score. However, improvements are small when comparing the BERT-tiny baseline to the smaller BioBERT models, where BERT-tiny outperforms both in recall. Over all six tested methods, the incremental differences in accuracy are also small, with only a 1.06% improvement in accuracy with the best performing model (BioBERT) over the best performing baseline (BERT-tiny). Among the BioBERT variants, the difference in improvement as model size increases is primarily seen through the decrease in false negatives and increase in true positives. Relevant tables and figures are shown below.

Method	Accuracy	Precision	Recall	F1
DistMult	0.915	0.921	0.928	0.925
TransE	0.921	0.914	0.935	0.925
LMKE$_{\texttt{BERT-tiny}}$	0.937	0.906	0.975	0.939
LMKE$_{\texttt{BioBERT}}$	**0.947**	**0.921**	**0.979**	**0.949**
LMKE$_{\texttt{TinyBioBERT}}$	0.940	0.917	0.968	0.942
LMKE$_{\texttt{biobert-tiny}}$	0.938	0.921	0.959	0.939

Table 2: Performance of baselines compared with our 3 LMKE configurations



(a) BioBERT (b) TinyBioBERT (c) biobert-tiny

Figure 2: Confusion Matrices for each BioBERT variant model

6 Analysis

6.1 Evaluating modifications to LMKE

In this project, we make three main modifications to LMKE: the use of textual descriptions, the use of support scores as labels during training, and the integration of Biomedical language models instead of general language models. Here, we analyze different impacts of each modification on triple classification accuracy.

We observe improvements in triple classification accuracy when the triple is passed along with descriptions. This makes sense, as it allows the language model to contextualize the entities with their descriptions, as well as the descriptions of the other entity or relation in the triple. We would expect further improvements with more complete textual descriptions for entities and relations.

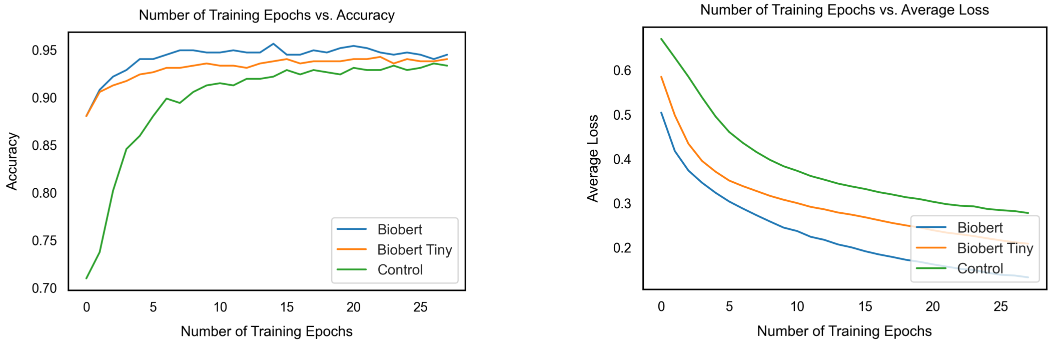
In testing the effect of support scoring, it is interesting to note that support scoring does not seem to have an effect on the accuracy of the model. This was consistent in all variants of BioBERT and the BERT-tiny baseline. A plausible explanation for this is that the goal of support scoring is to quantify the ferquency of a certain interaction (triple) in the biomedical corpora. In other words, high support scores correlate to more frequent mentions in PubMed. However, because we leverage language models in LMKE, it is possible that the frequency of the interaction in biomedical literature was already learned by the model while pretraining since BioBERT saw triples with high support scores much more frequently in pretraining than triples with low support scores

We see that increasing the size of the language model used by LMKE increases the accuracy and F1 score of the model. This is seen by incremental improvements in accuracy from biobert-tiny, the smallest model, to TinyBioBERT, and finally with BioBERT. However, the improvments in accuracy are relatively small, with less than 1% improvement between BioBERT and biobert-tiny. Several possible explanations regarding the differences in convergence during training and the size of our validation and test sets are analyzed below. Interestingly, this seems to align with previous tests by

Wang et al. (2022), which found BERT-tiny performance to be comparable, if not better, than BERT performance on general knowledge graph completion tasks.

6.2 Convergence differences between language models

One explanation for the small difference in accuracy between BioBERT and smaller models is the rate of convergence in validation accuracy during training.



(a) Accuracy plot

(b) Loss plot

Figure 3: Accuracy and loss plots side by side

We observe an significantly higher validation accuracy when using biomedical language models for the first 5 epochs when comparing against the baseline BERT-tiny model. This supports the notion that biomedical language models have better a better representation of the entities in the dataset. However, we see that BERT-tiny is able to converge to similar performance to the two biomedical language models even though it is smaller than BioBERT and is not pretrained on biomedical corpora. This suggests that though biomedical language models have an advantage through better pretrained representations, general purpose language models can quickly learn representations of the entities from the structure of the knowledge graph. This may also indicate that biomedical knowledge graph embeddings are still primarily determined by structure, and not descriptions. In addition, this could explain why size of the model seems to have a small role in classification performance, because the size of the model primarily benefits the model through better representation of the text of the triples, not the structure. However, it is clear that descriptions are still helpful, as including them resulted in improved accuracy across all LMKE variants.

6.3 Validation and test size as a performance factor

Another possible explanation for the relatively small differences in testing and validation is the small size of both the sets used. Because of the sparsity of rare disease treatments in GNBR, there were less than 2000 points in the validation and test set combined. Increasing the size of the validation and test sets would likely improve the resolution between the accuracies of different models. This is especially true in the validation set, which determined the stopping point and model state after training. It is likely that the small size of the validation set was not representative of the knowledge graph and was a primary contributing factor to the plateaued accuracies seen in Figure 3. Further work and more compute would allow us to bypass this issue by creating and testing a larger dataset.

7 Conclusion and Future Work

We show that leveraging biomedical language models to generate knowledge graph embeddings improves on SOTA methods for knowledge graph completion in the context of biomedical networks. By framing our problem as a triple classification task, we find that using BioBERT with SOTA knowledge graph embedding methods that leverage language models has promise in drug repurposing tasks. We also provide evidence that knowledge graph embeddings are primarily determined by the structure of the knowledge graph. However, improvements when adding textual descriptions indicate that while small, methods that leverage descriptions do improve on structure based methods.

The primary limitation to this project was computational power. This constraint forced us to work with a smaller dataset, pretrain a smaller BioBERT model for the project, and focus on triple classification as a representative task for drug repurposing. When comparing to benchmark sets like PharmKG, our dataset had a ten fold decrease in the average degree of each entity. While some of this is explained by the sparsity of rare disease interactions in biomedical literature, it is also a byproduct of a smaller dataset. With more compute, we can mitigate the drop in performance caused by sparsity, and create larger validation and test sets for future experiments.

While this project framed the drug repurposing task as a triple classification problem, it is perhaps more useful to think of the drug repurposing problem as one where models can find and rank new treatments, not just validate ones that are proposed to the model. The former task is a high level explanation of link prediction, where models are able to rank all possible entities given a certain relation and entity. Important future work should focus on link prediction to enable models to suggest top drug candidates for certain diseases. Due to our time and computational resource constraint, we were unable to test link prediction, which takes considerably more resources because it ranks all possible entities for each train and test triple. Practically, future work should also focus on predicting new use cases for drugs and applying this model to several rare diseases to suggest top candidates for computational and wet lab testing.

8 Contributions

All collaborators contributed equally in developing the research objective, conducting literature review, and executing the project. Yash Patil has additional contributions in pretraining biobert-tiny, running baseline metrics, and analysis of convergence. John Wang has additional contributions in modifications to LMKE, generating the dataset, running experiments for BioBERT variants and baselines with LMKE, creating scripts for evaluation and analysis, and setting up workflow in Sherlock.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67.

Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. 2019. Embedding uncertain knowledge graphs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoonbee Kim, Yi-Sue Jung, Jong-Hoon Park, Seon-Jun Kim, and Young-Rae Cho. 2022. Drug-disease association prediction using heterogeneous networks for computational drug repositioning. *Biomolecules*, 12(10).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Bethany Percha and Russ B Altman. 2015. Learning the structure of biomedical relationships from unstructured text. *PLoS computational biology*, 11(7):e1004216.

Bethany L Percha and Russ B. Altman. 2018. A global network of biomedical relationships derived from text. *Bioinformatics*, 34:2614 – 2624.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2020. Kepler: A unified model for knowledge embedding and pre-trained language representation.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022. Language models as knowledge embeddings.

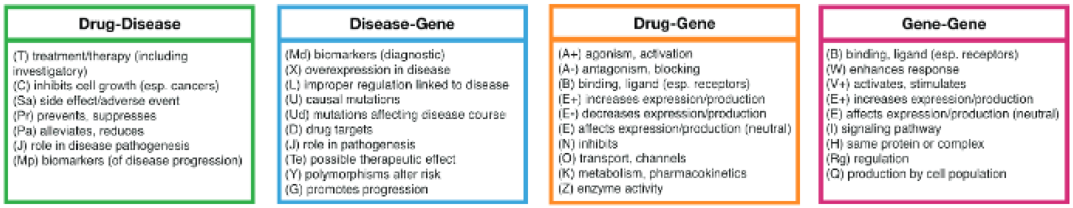
Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion.

Shuangjia Zheng, Jiahua Rao, Ying Song, Jixian Zhang, Xianglu Xiao, Evandro Fei Fang, Yuedong Yang, and Zhangming Niu. 2020. PharmKG: a dedicated knowledge graph benchmark for bomedical data mining. *Briefings in Bioinformatics*, 22(4). Bbaa344.

9 Appendix



Drug-Disease	Disease-Gene	Drug-Gene	Gene-Gene
(T) treatment/therapy (including investigatory) (C) inhibits cell growth (esp. cancers) (Sa) side effect/adverse event (Pr) prevents, suppresses (Pa) alleviates, reduces (J) role in disease pathogenesis (Mp) biomarkers (of disease progression)	(Md) biomarkers (diagnostic) (X) overexpression in disease (L) improper regulation linked to disease (U) causal mutations (Ud) mutations affecting disease course (D) drug targets (J) role in pathogenesis (Te) possible therapeutic effect (Y) polymorphisms alter risk (G) promotes progression	(A+) agonism, activation (A-) antagonism, blocking (B) binding, ligand (esp. receptors) (E+) increases expression/production (E-) decreases expression/production (E) affects expression/production (neutral) (N) inhibits (O) transport, channels (K) metabolism, pharmacokinetics (Z) enzyme activity	(B) binding, ligand (esp. receptors) (W) enhances response (V+) activates, stimulates (E+) increases expression/production (E) affects expression/production (neutral) (I) signaling pathway (H) same protein or complex (Rg) regulation (Q) production by cell population

Figure A.1: Relation types in GNBR

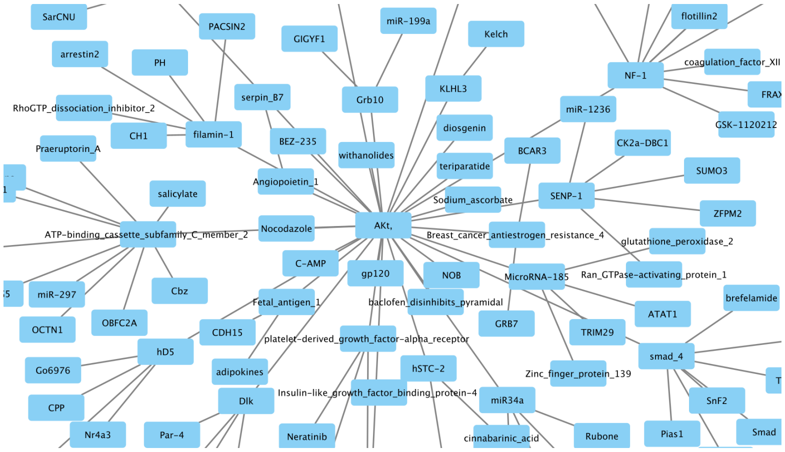


Figure A.2: Visualization of a small area of the dataset without relation types shown