# Drug Re-positioning via Text Augmented Knowledge Graph Embeddings

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Drug re-positioning, modeled as a link prediction problem over medical knowledge graphs (KG), has great potential in finding new usage or targets for approved medicine with relatively low cost. However, the semantic information in medical KGs is rarely utilized, let alone the external medical databases curated by domain experts. This work attempts to integrate textual descriptions of biomedical KG entities in training knowledge graph embeddings (KGE) and evaluates its effectiveness for drug re-positioning. We implement multiple text augmentation methods on TransE as a case study and further apply the best method on other embedding models. Both qualitative and quantitative error analyses with two novel metrics are conducted to shed light on the effects of adding textual information in our model. We conclude that textual information is generally useful, but it may also backfire.

## 1 Introduction

Drug re-positioning identifies new treatments for diseases from the developed and approved drugs to save costs and time. With this goal, using medical knowledge graphs (KGs) for drug re-positioning has attracted much attention for its efficiency in the target discovery stage (Nam et al., 2019, 2020; Sang et al., 2018; Sosa et al., 2019; Kanatsoulis and Sidiropoulos, 2021). Unlike crowdsourced KGs, medical KGs typically contain descriptions of medical entities from reliable sources that are essential literature references to assist doctors in comprehending and discovering new treatments. Moreover, prior works have shown performance gain in the general KG domain with literal augmentation with numeracy (Kristiadi et al., 2019) and entity descriptions (Zhong et al., 2015). Feeding such texts to KG embedding models to address drug-repositioning is expected to yield benefits yet remain underexplored. In this **work-in-progress** paper, we propose methods to incorporate textual information about medical entities into KG embedding models and investigate the effects by two novel evaluation metrics. The experiments use different techniques for incorporating texts on TransE model with Drugbank (Wishart et al., 2006) dataset, which improves much in Hits@K and MRR. We are encouraged to apply the most effective augmentation on other KG embedding approaches, where our current findings suggest that infusing text can be useful for some methods but surprisingly harmful in certain cases.

## 2 Methdology

**Task Formulation** A medical knowledge graph $G = (V, E)$ contains entities about genes, compounds, symptoms and etc., and relations describing activities among entities such as *treats* or *palliates*. A fact in $G$ is represented in a triple $< h, r, t >$ where $h, t \in V$ and $r \in E$. The drug re-positioning task is modeled as a link prediction task such that we evaluate tail predictions $< h, r, ? >$ for a subset of relations $E' \subset E$ related to drug treatment. Figure 1 illustrates the task formulation.

Submitted to NeurIPS 2021 AI for Science Workshop, Mentorship Track.

### 2.1 KGE with Text Augmentation

Given an entity in $G$, we denote $\mathbf{e}_s \in \mathbb{R}^d, \mathbf{e}_t \in \mathbb{R}^n$ as its KG structural embedding and textual embedding respectively. The final embedding $\mathbf{e} \in \mathbb{R}^d$ is trained for comparison with models without text. We specify three composition methods for fusing information from text and KG: **concatenation**, **addition** and **gating**, where detailed descriptions can be found in Appendix A.1. Based on these operations, we propose five model architectures shown in Figure 2. Model V0 and V1 utilize concatenation, model V2 and V3 addition, and model V4 gating.

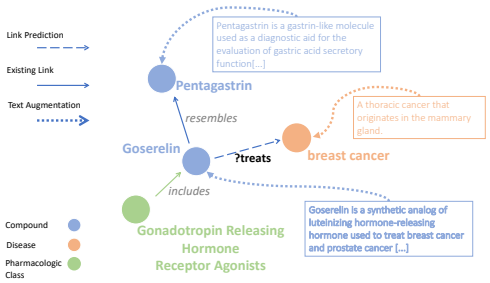


Figure 1: Illustration of drug re-positioning as KG link prediction. The available texts of entities are injected in training KG embeddings. Boxed texts are external descriptions, colored texts are entity labels and black-grey texts are relations.

### 2.2 Evaluation

We propose two new metrics, *% of Disease @ K* and *Unique Entity @ 1*, to shed some light on model errors. *% of Disease @ K* refers to the percentage of predicted tails being Disease among the top N predictions. This metric provides an interpretable way of checks how well the model works in drug repurposing - a prediction to non-disease entity type for drug treatment would be erroneous. When $K = 1$, this metric is an aggregated precision measure over all possible diseases and thus different from Hits@K which can be seen as a recall measure. Moreover, *Unique Entity @ 1* refers to the total amount of unique entity predictions the model makes. This metric to some extent infers disease diversity in predictions of drug treatment that the model can make.

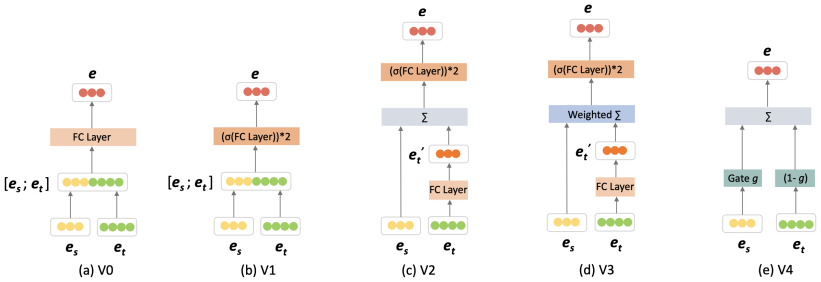


Figure 2: Model configurations for textual augmentation. In (a), we show V0 where we concatenate the textual embedding and the KG embedding after which we transform into a lower-dimensional embedding. In (b), we show V1 which adds a fully connected layer with activation on top of V0. In (c), we show V2 which uses addition to fuse information. In (d), we show V3 which performs a weighted sum when we fuse the two inputs. In (e), we apply a gating function in V4 which is based on both the textual and KG embeddings and fuses the two inputs.

## 3 Experiments

### 3.1 Data

Hetionet (Himmelstein et al., 2017) is a benchmark KG for drug discovery that contains 45,158 entities and 24 different relation types, where 17,345 available descriptions are scraped[1]. We only use triples with relation type "Compounds treats Disease" (CtD) in evaluation. Following the setting of Liu et al. (2021), we acquire a training set containing 483 "CtD" triples and all other non-CtD triples, a validation set of "CtD" 121 triples, and a test set of 151 "CtD" triples (see Appendix A.3).

[1]The example texts and statistics of each entity type can be found in Appendix A.2

| Model | Hits@1 | Hits@3 | Hits@5 | Hits@10 | MRR |
|---|---|---|---|---|---|
| TransE | 0.00662 | 0.07947 | 0.1457 | 0.29800 | 0.09928 |
| TransETextV0 | **0.05960** | **0.17220** | **0.23840** | **0.36420** | **0.16120** |
| TransETextV1 | 0.03311 | 0.08609 | 0.12580 | 0.18540 | 0.09137 |
| TransETextV2 | 0.01987 | 0.07285 | 0.09934 | 0.18540 | 0.08760 |
| TransETextV3 | 0.01987 | 0.09272 | 0.16560 | 0.26490 | 0.10820 |
| TransETextV4 | 0.01987 | 0.09934 | 0.13250 | 0.25170 | 0.10580 |

Table 1: Effect of different textual augmentation methods on TransE

| | Model | Hits@1 | Hits@3 | Hits@5 | Hits@10 | MRR |
|---|---|---|---|---|---|---|
| Baselines | TransE | 0.00662 | 0.07947 | 0.14570 | 0.29800 | 0.09928 |
| | DistMult | 0.10600 | 0.19870 | 0.28480 | 0.45030 | 0.20800 |
| | ProjE | 0.01987 | 0.05960 | 0.12580 | 0.25830 | 0.08486 |
| | RotatE | 0.17880 | 0.33110 | 0.39070 | 0.53640 | 0.29640 |
| Text Augmented Models | TransEText | 0.05960 | 0.17220 | 0.23840 | 0.36420 | 0.16120 |
| | DistMultText | 0.07285 | 0.19210 | 0.31130 | 0.45700 | 0.18720 |
| | ProjEText | 0.01325 | 0.09272 | 0.15890 | 0.26490 | 0.09373 |
| | RotatEText | 0.03974 | 0.05960 | 0.08609 | 0.2053 | 0.09284 |

Table 2: Effect of the concatenation textual augmentation method on different baselines. Baselines and text augmented models share the same training settings.

### 3.2 Training

For baselines, we consider embedding-based methods TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ProjE (Shi and Weninger, 2017), and RotatE (Sun et al., 2019), where each is trained to minimize the margin ranking loss with Adagrad optimizer (Duchi et al., 2011). The uniform negative sampling strategy is used.[2] The text embeddings of entity descriptions are encoded with BioBert (Lee et al., 2020) model, a pre-trained model on biomedical corpora, capped at 512 tokens. For entities without description, we use a zero vector as a placeholder. All models are implemented with the Pykeen[3] library (Ali et al., 2021) and training configurations are described in Appendix A.4

## 4 Results

### 4.1 Case Study on TransE

We apply all five text augmentation models on TransE (see Table 1) and they all provide performance improvements on TransE. The simple linear transformation using concatenation (model V0) gives the best performance, which is encouraging for gains in performance and efficiency. The method is thus extended to other embedding models to understand its generality. In addition, we analyze the effect of the amount of text used (see Appendix A.5) that more text indeed boosts better performance.

### 4.2 Comparison of Baselines and Text Augmented Models

In Table 2, we compare the performance of four baselines and their counterparts using model V0. TransE improves the most after adding text, followed by ProjE and DistMult. The performance gain increases along with less complex baseline models. However, baseline RotatE performs best out of all models with and without textual enhancement. Intriguingly, it suffers much from text augmentation, possibly resulted from the complex model of RotatE itself or unknown conflicts between RotatE embedding operation and text incorporation approach.

### 4.3 Quantitative Error Analysis

While text augmentation works well in general KG domain, the improvement of TransEText and the degraded performance of RotatEText raise cautions of adopting same text enhancement in our high-stake settings. To better understand the predictions, we evaluate the models using our new metrics for drug-repositioning in Table 3. To begin with, TransEText doubles *% of Disease @ 1* of TransE, which indicates external texts guide TransE to make less obvious mistakes (any predictions

[2]More detailed settings are listed in the Appendix A.4.

[3]`https://pykeen.readthedocs.io/en/stable/#`

do not belong to Disease are completely wrong under our settings). In contrast, RotatE with text results in a much smaller *% of Disease @ 1*. This metric help understand the opposite behaviors of TransEText and RotatEText compared with their vanilla models. Besides, text augmentation seems to drive *% of Disease @ K* to a similar range(83%), except for RotatEText. We also observe that the *% of Disease@K* well correlates with Hits@K and MRR on baselines as well as on the text augmented models shown in Table 2 and 3. In terms of *Unique Entities @1*, the values vary widely on the baseline methods[4]. RotatE achieves the smallest and even lower than the ground truth, which suggests less diversity in predictions. Furthermore, the text augmentation mostly reduces *Unique Entities @1* from baselines, such that the model may suggest a smaller range of diseases in inference. Such conservative pattern is likely resulted from inputting the same text on distinct KGE structures.

| | Model | % of Disease @1 | % of Disease @10 | Unique Entities @1 |
|---|---|---|---|---|
| | Ground Truth | 100% | N/A | 52 |
| Baselines | TransE | 35.76% | 87.09% | 113 |
| | DistMult | 98.01% | 98.01% | 44 |
| | ProjE | 80.13% | 72.78% | 76 |
| | RotatE | 98.67% | 98.08% | 38 |
| Text Augmented Models | TransEText | 79.47% | 82.91% | 74 |
| | DistMultText | 87.41% | 85.30% | 53 |
| | ProjEText | 83.44% | 84.64% | 71 |
| | RotatEText | 68.22% | 53.44% | 36 |

Table 3: Quantitative Error Analysis of the model prediction. We show the result of the baseline methods as well as text augmented methods in two metrics on testing set: % of Disease @ K and Unique Entities @ 1. The former refers to the percentage of predicted tail entities belonging to the class Disease. The latter refers to the total number of unique tail entity prediction, regardless of the input.

### 4.4 Qualitative Error Analysis

Lastly, to further shed some light on the effects of text on RotatE, we carefully examine model predictions and provide a negative example in Table 4:

| Head | **Tail(Truth)** | **Tail(Incorrect Prediction)** |
|---|---|---|
| Guanfacine [Guanfacine is an alpha-2A adrenergic receptor agonist used to treat ADHD] | Hypertension [An artery disease characterized by chronic elevated blood pressure in the arteries.] | Panic Disorder [An anxiety disorder that is characterized by unexpected and repeated episodes of intense fear accompanied by physical symptoms that may include chest pain, heart palpitations, shortness of breath, dizziness, or abdominal distress] |

Table 4: A negative example of RotatEText

The incorrect prediction of RotatEText seems plausible since ADHD and panic disorder are both mental disorders if solely based on text descriptions. Complex biomedical relationships captured by baseline RotatE may lose out as RotatEText learns superficial patterns from text in this scenario. Meanwhile, for all other baseline KGE methods, this is less of an issue because they were less capable of modeling complexity of biomedical process so that the benefits of text augmentation outweigh the harm. Still, even within RotatEText, there are cases in test set demonstrating the opposite situation, where textual information directs RotatEText to the right prediction that RotatE predicts wrong. Whether or not the performance of a KGE model benefits from textual information comes down to how many samples in the dataset can be predicted easier with textual data alone versus how many can be predicted with KG data alone.

## 5 Future Work

Despite abovementioned examination of erroneous samples, we would like to build a systematic way to visualize embedding projections with and without text. The current negative results from RotatEText additionally call for suitable ways to infuse textual information with methods using attention-based mechanisms. To scale up the experiments, we also find it necessary to extend the work on larger datasets such as OpenBioLink (Breit et al., 2020).

[4]For reference, we include the *Unique Entity @ 1* for the ground truth.

## References


Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22(82):1–6.

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, and William L Hamilton. 2021. Understanding the performance of knowledge graph embeddings in drug discovery. *arXiv preprint arXiv:2105.10488*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9.

Anna Breit, Simon Ott, Asan Agibetov, and Matthias Samwald. 2020. Openbiolink: a benchmarking framework for large-scale biomedical link prediction. *Bioinformatics*, 36(13):4097–4098.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. 2017. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife*, 6:e26726.

Charilaos I Kanatsoulis and Nicholas D Sidiropoulos. 2021. Tex-graph: Coupled tensor-matrix knowledge-graph embedding for covid-19 drug repurposing. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 603–611. SIAM.

Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. 2019. Incorporating literals into knowledge graph embeddings.

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.

Yushan Liu, Marcel Hildebrandt, Mitchell Joblin, Martin Ringsquandl, Rime Raissouni, and Volker Tresp. 2021. Neural multi-hop reasoning with logical rules on biomedical knowledge graphs. In *European Semantic Web Conference*, pages 375–391. Springer.

Yonghyun Nam, Myungjun Kim, Hang-Seok Chang, and Hyunjung Shin. 2019. Drug repurposing with network reinforcement. *BMC bioinformatics*, 20(13):1–10.

Yonghyun Nam, Jae-Seung Yun, Seung Mi Lee, Ji Won Park, Ziqi Chen, Brian Lee, Anurag Verma, Xia Ning, Li Shen, and Dokyoon Kim. 2020. Network reinforcement driven drug repurposing for covid-19 by exploiting disease-gene-drug associations. *arXiv preprint arXiv:2008.05377*.

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*, 7:8404–8415.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Daniel N Sosa, Alexander Derry, Margaret Guo, Eric Wei, Connor Brinton, and Russ B Altman. 2019. A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*, pages 463–474. World Scientific.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. 2006. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272, Lisbon, Portugal. Association for Computational Linguistics.

# A Appendix

## A.1 Description of Information Fusion Operations

**Concatenation** An efficient combination is to concatenate the structural and textual embeddings and project the result embedding on $\mathbb{R}^d$ such that, $\mathbf{e} = f([\mathbf{e}_s; \mathbf{e}_t])$, where $f$ is some mapping reduce dimension to $d$.

**Addition** Furthermore, we propose additive composition to infuse the text information such that, $\mathbf{e} = \mathbf{e}_s + \sigma(\mathbf{e}_t)$, where $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is typically some function[5] to map the textual embedding to the structural embedding size. We investigate two strategies where $\sigma$ is simply linear function and is non-linear activated on linear transformation.

**Gating** The gating mechanism defines a gate parameter $\mathbf{g} \in \mathbb{R}^d$ to control information flow from $\mathbf{e}_s$ and $\mathbf{e}_t$ such that,$\mathbf{e} = \mathbf{g} \odot \mathbf{e}_s + (1 - \mathbf{g}) \odot \mathbf{e}_d$, where $\odot$ refers to element-wise multiplication.

## A.2 Hetionet Text

In this section, we display supplementary statistics for the text descriptions used in our model in Table 5.

| | Counts | Example |
|---|---|---|
| Anatomy | 400 | A nerve which runs near the ulna bone. |
| Biological Process | 11,041 | The chemical reactions and pathways resulting in the formation of eye pigments, any general or particular coloring matter in living organisms, found or utilized in the eye. |
| Cellular Component | 1,364 | The side (leaflet) of the early endosome membrane that faces the cytoplasm. |
| Compound | 1,298 | Bromhexine is a mucolytic drug used to decrease the viscosity of mucus in the airway, enhancing mucus clearance. |
| Disease | 124 | A demyelinating disease that involves damage to the fatty myelin sheaths around the axons of the brain and spinal cord resulting in demyelination and scarring. |
| Molecular Function | 2,749 | Catalysis of the reaction: alkene-CoA + H2O = alcohol-CoA. Substrates are crotonoyl-CoA (producing 3-hydroxyacyl-CoA) and 2,3-didehydro-pimeloyl-CoA (producing 3-hydroxypimeloyl-CoA) |
| Pathway | 283 | Model of hypoxia mediated EMT and stemness. |
| Pharmacologic Class | 86 | Compounds with a benzene ring fused to a thiazole ring. |

Table 5: Statistics of text descriptions of Hetionet dataset used in final KG embedding training. Entity types “gene", “side effect" and “symptoms" are not included due to some inaccessibility.

## A.3 Hetionet Statistics and Splits

The statistics of Hetionet and the training, validation, testing splits can be seen in Table 6.

| | Triples | CtD Triples |
|---|---|---|
| Train | 2,249,925 | 483 |
| Validation | 121 | 121 |
| Test | 151 | 151 |

Table 6: Statistics of Hetionet Dataset and our train, validation test division

## A.4 Baseline Hyperparameter Configurations

For hyperparameters of TransE, DistMult and RotatE, we referred to Bonner et al. (2021) and applied the mean value of the top 5 configurations through their HyperParameter Optimisation (HPO) Akiba

[5]complex space might sometimes involve

et al. (2019) experiments on Hetionet. The settings for ProjE are the best configuration of our 10-trial HPO experiments.

| Model | **Emb Size** | **Num Epochs** | **Learning Rate** | **Num Neg** |
|---|---|---|---|---|
| **TransE** | 285 | 580 | 0.022 | 49 |
| **DistMult** | 214 | 400 | 0.030 | 61 |
| **ProjE** | 200 | 800 | 0.050 | 10 |
| **RotatE** | 483 | 840 | 0.028 | 31 |

Table 7: Baseline Hyperparameter Profile

### A.5 Impact of the Amount of Texual Information

The impact of the amount of text in model performance is investigated in this section. We train the TransETextV0 model with a much smaller amount of text of 1,533 textual descriptions of the "Compound" class. We name this model **TransETextV0 - Compound**. All other settings remain unchanged. The result is shown in Table 1. We see that the performance drops to a large degree, on all metrics. At the same time, TransETextV0 - Compound still outperforms the vanilla TransE on all metrics but Hits@10, showcasing the power of incorporating textual information into KG methods.

We also explore the impact of using a pretrained KG model as initialization for the joint text-KG embedding model. Specifically, we initialize the KG part of the TransETextV0 model with the pretrained TransE model. We name the new model **TransETextV0 - Pretrained**. As part of the model is already trained, we reduce the learning rate to 0.001. All other parameters remain the same. The performance of this model, unfortunately, is subpar and in some metrics even worse than the vanilla TransE model.