Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 2 April 2015
Manuscript Category

OXFORD

DOUBLER: Unified Representation Learning of Biological Entities and Documents for Predicting Protein–Disease Relationships

Timo Sztyler [1,*] and Brandon Malone [1]

[1] NEC Laboratories Europe, Heidelberg, 69115, Germany

[*] To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: We propose a system that learns consistent representations of biological entities, such as proteins and diseases, based on a knowledge graph and additional data modalities, like structured annotations and free text describing the entities. In contrast to similar approaches, we explicitly incorporate the consistency of the representations into the learning process. In particular, we use these representations to identify novel proteins associated with diseases; these novel relationships could be used to prioritize protein targets for new drugs.
Results: We show that our approach outperforms state-of-the-art link prediction algorithms for predicting unknown protein–disease associations. Detailed analysis demonstrates that our approach is most beneficial when additional data modalities, such as free text, are informative.
Availability: Code and data are available at: `https://github.com/nle-sztyler/research-doubler`
Contact: timo.sztyler@neclab.eu

1 Introduction

Biological processes within the human body are the result of complex interactions among various molecules, such as DNA, RNA, proteins, metabolites, enzymes, and more. Disease phenotypes result from the disruption of these processes; likewise, effective treatments often aim to restore these processes to their normal, healthy state. Thus, it is imperative to accurately and thoroughly characterize the relationships among the molecules, diseases, and other entities involved in these processes.

Traditional approaches for characterizing these relationships entail expensive and time-consuming biological wet lab experiments. Due to the cost of these experiments, existing datasets are highly incomplete (Härtner *et al.*, 2018; Madeddu *et al.*, 2019). Thus, while this experimental approach remains the gold standard for determining their truth, it is important to devise more cost-effective methods to characterize these relationships.

Increasing evidence demonstrates that artificial intelligence (AI) methods are also very effective for identifying novel relationships (Ghiassian *et al.*, 2015; Caldera *et al.*, 2017; Härtner *et al.*, 2018). Of course, high-confidence relationships predicted by the AI methods can (and should) still be validated with traditional wet lab approaches.

In this paper, we present an approach for learning novel relationships among the various biological entities. We adopt a network-based approach in which each entity is represented as a node in a graph, and each relationship is represented as an edge between the respective entities; different relationship types are represented with different edge types. We refer to this network as a knowledge graph (KG). Learning novel relationships is then equivalent to predicting "missing" edges from the KG (Sharma *et al.*, 2015; Caldera *et al.*, 2017; Vlaic *et al.*, 2018).

Similar network-based approaches have been proposed to address this problem before (Caldera *et al.*, 2017; Madeddu *et al.*, 2019; Malone *et al.*, 2018; Nováček and Mohamed, 2020). The existing approaches can be grouped into three categories (Agrawal *et al.*, 2018): neighborhood-based, diffusion-based, and representation-based. These rely on different topological properties of the network, such as motifs, communities, hubs, clusters, node centrality, shortest paths, and others to predict new edges. Further, some methods also account for biological features of the nodes. For example, proteins may be annotated with their functions (Asif *et al.*, 2018), text from academic papers in which they are described (Zhou and Fu, 2018), or patient sequencing data (Luo

© The Author 2020.

et al., 2017); we refer to this additional information as a *document* for the respective entity.

The representation-based approaches are the most similar to ours. These approaches use neural networks to learn *embeddings*, or real-valued numeric vectors, such that entities which are close in the graph and have similar relationships also have similar embeddings. A machine learning model is then trained to predict novel relationships based on the embeddings.

Currently, the proposed approaches do not jointly account for the KG structure and the associated documents in a consistent manner (Agrawal *et al.*, 2018; Vlaic *et al.*, 2018; Ata *et al.*, 2018). In particular, they do not share information between the KG structure and documents about the entities in a unified manner.

We address this shortcoming with a novel neural network learning strategy, DOUBLER (**DO**cuments **U**nified with **B**iological networks for **LE**arning **R**epresentations), which combines the KG structure with the documents in a unified framework in which we force both embeddings to be similar. The main advantage compared to the obvious approach (learning independent embeddings and concatenating them) is that this "pulls together" the embeddings of entities which are distant in the KG but have similar documents; likewise, it "pushes apart" embeddings of genes that are close in the KG but have very different documents. Thus, novel relationships are predicted using a unified view of the KG and documents.

In order to demonstrate the efficacy of our approach, we construct a KG consisting of proteins and diseases. We then compare our approach to existing state-of-the-art bioinformatics and machine learning algorithms for predicting novel protein–disease associations. Empirically, we show that our approach outperforms the competing methods. More detailed analysis demonstrates that our approach is particularly beneficial for cases in which the documents for proteins are informative.

In Section 2, we describe the datasets used to construct the disease–gene network, while Section 3 presents our new approach for learning consistent representations based on documents and the network. The results and discussion are given in Section 4, and Section 5 concludes the paper.

2 Materials

DOUBLER requires two data sources: (1) the knowledge graph (KG) and (2) the documents describing entities in the KG.

2.1 Biomedical Knowledge Graph

A knowledge graph $G = (V, E)$ consists of a set of nodes V and a set of edges E. Each node in the KG corresponds to one entity, such as a protein or disease. The edges characterize the known relationships between these entities.

As an example of constructing the knowledge graph G in order to demonstrate and evaluate the proposed approach, we extracted entities and relationships from the following sources.

- `disease - associated_with - protein`
  DisGeNET: Lo Surdo *et al.* (2017)
- `protein - interacts_with - protein`
  STRING: Szklarczyk *et al.* (2018)

Additionally, we created `disease - similar_to - disease` relationships using an existing approach (Gottlieb *et al.*, 2011). Further, some data sources provided confidence values for the relationships. We took these into account to filter relationships with a low reliability since those are already predictions.

Our KG G includes two types of entities: diseases and proteins. It includes three relationship (edge) types. We use e^t to refer to edge type t. In the following, the facts which are encoded by the graph in G are represented as a set of triples of the form (h, r, t) where $h \in V$ and $t \in V$ are the head and tail entities and $r \in E$ is a relation type.

As an example, Figure 1 illustrates an excerpt of G. It shows that the disease "Asthma" (D1) is associated with "Gene 108" (P2) and "Gene 142" (P1). From a practical point of view, it is expensive to identify novel relations and most of them are still unknown.

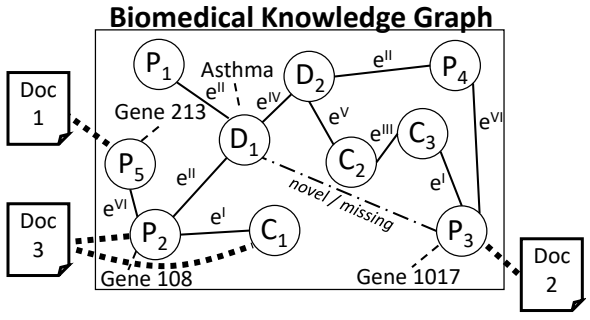

Fig. 1: Composition the biomedical knowledge graph (excerpt).

Recently, a variety of similar biomedical KGs have been proposed, such as (Breit *et al.*, 2020) and (Mohamed *et al.*, 2020). Due to the similarity of the source databases, we do not further consider those here; nevertheless, DOUBLER can be directly applied to those KGs, as well.

2.2 Mining Biomedical Documents

The documents describe the related entities in the KG in more detail. In particular, we use the following sources and associated types of information.

- Gene Ontology (Gene Ontology Consortium, 2004): protein functions and locations in the cell (structured annotations)
- Human Phenotype Ontology (Köhler *et al.*, 2014): phenotypes associated with diseases (structured annotations)
- MyGene.info (Wu *et al.*, 2013): protein descriptions (free text)
- DisGeNET (Lo Surdo *et al.*, 2017): disease mechanisms and associated mutations (free text)

A common vocabulary is created based on all documents by applying Subword Neural Machine Translation (Sennrich *et al.*, 2016). To reduce vocabulary size, the method breaks non-frequent words into subword units. This is done in an unsupervised manner. As a result, it is possible to encode any (rare) word in subword units. Thus, the vocabulary size is fixed, and we avoid problems from out-of-vocabulary tokens.

Figure 1 illustrates that each document is associated with at least one of the entities from the KG. A document can be associated with more than one entity, and each entity can be associated with more than one document. We assign

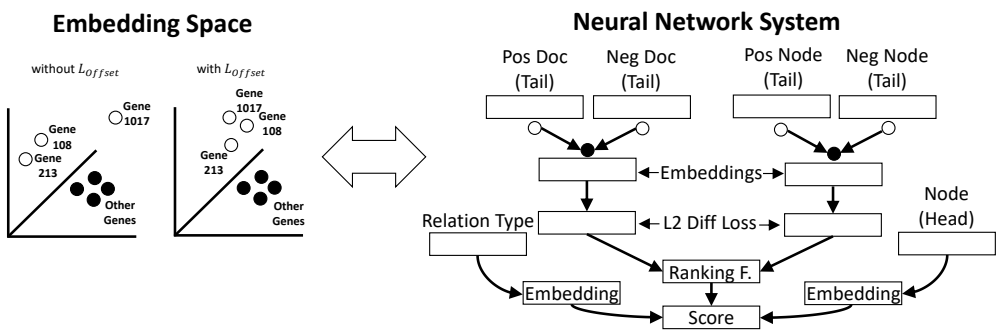

Fig. 2: Conceptual connection illustrating how the joined learning of the different modality (right) affects the embedding space (left).

documents to entities based on existing assignments in the source databases.

Figure 1 illustrates the connection between documents and proteins. More formally, necessary data for DOUBLER is a set of biomedical documents $\mathbb{D}$, where each $d \in \mathbb{D}$ is a multiset of tokens from the vocabulary. Further, $\forall d \in \mathbb{D}$, it holds that $\exists (v, d) \in (V, \mathbb{D})$ where $(V, \mathbb{D})$ is a set of pairs. We denote this as $G_B = ((V, E), (V, \mathbb{D}))$. G_B will be used as input for our neural network.

3 Methods

We now describe a novel neural network approach for learning to predict novel relationships which combines the knowledge graph (KG) G_B and the related set of biomedical documents $\mathbb{D}$ in a unified framework, visualized in Figure 2.

3.1 Training

We refer in the following to "proteins" and "diseases" as possible head and tail entities where the corresponding relation is "`associated_with`" ($e^{d_aw_p}$). For training the machine learning model, we keep a particular disease (e.g., "Cardiomyopathy") and the relation r_k (e.g., "`associated_with`") constant as the head h and relation type r, respectively, and group the proteins into V_p^+ and V_p^- where $\forall v_p^+ \in V_p^+$ it holds that a triple in the form of (h, `associated_with`, v_p^+) exists in KG. It holds that $V_p^+ \cap V_p^- = \emptyset$, $V_p^+ \cup V_p^- \subseteq V_p$ and $V_p \subseteq V$. Analogously, we have $\mathbb{D}_p^+$ and $\mathbb{D}_p^-$ which describe the corresponding biomedical documents of the elements in V_p^+ and V_p^-. That is, every protein in V_p^+ is a correct tail for the triple and is associated with the disease, while every protein in V_p^- is an incorrect tail and is not known to be associated with the disease. As indicated in Figure 3, we conceptually train our machine learning model with a single sample at a time. That is, we choose $v_j \in V_p^+$ and the corresponding document $d_j \in \mathbb{D}_p^+$. As a first step, this sample is embedded, where the embedding of the nodes V and the documents $\mathbb{D}$ happens independently:

$$Emb_U(x_j) = \frac{1}{\|x_j\|} \sum_{i \in x_j} \Theta_i^{Emb_U} \tag{1}$$

where $U \in \{V, \mathbb{D}\}$, Emb_U is the respective embedding function, $\Theta_i^{(Emb_U)}$ is the embedding of element i, x_j is the corresponding element (that is, v_j or d_j), $i = \|x_j\|$. The cardinality for d_j is defined by the number of words in the related documents. The cardinality for v_j is always 1; it is an indicator for the entity in the graph. Second, we apply negative sampling (Kotnis and Nastase, 2018) to randomly select V_p^- and $\mathbb{D}_p^-$, where $V_s^- \subseteq V_p^-$ and $\mathbb{D}_s^- \subseteq \mathbb{D}_p^-$. That is, these sets contain incorrect tails for the selected head and relation. Analogously, we embed the set of negative elements as follows:

$$Emb_U^{NEG}(Y_s^-) = \frac{1}{\|Y_s^-\|} \sum_{y_s^- \in Y_s^-} Emb_U(y_s^-) \tag{2}$$

where Y_s^- is the set of negative samples (that is, V_s^- or $\mathbb{D}_s^-$). The cardinalities of both V_s^- and $\mathbb{D}_s^-$ are equal to the number of negative samples. The number of negative samples is a hyperparameter of the method. The embedding of the relation is defined as follows:

$$Emb^R(r_k) = \Theta_k^{Emb^R} \tag{3}$$

where k is the index of a particular relationship type. More concretely, Emb_V is a neural network which gives an embedding for each entity in the KG, while Emb^R is a neural network which gives an embedding for each relationship type. Additionally, $Emb_{\mathbb{D}}$ is a neural network which gives an embedding for each token in the vocabulary, and the embedding for a document is simply the average embedding of all of its tokens. Emb_U^{NEG} is a simple function that gives the average embedding of all entities passed to it.

3.2 Learning unified embeddings

The neural network calculates the Euclidean distance between the embedding of the positive sample x_j and the center of the negative samples Y_s^-:

$$distance_U(x_j, Y_s^-) = \|Emb_U(x_j) - Emb_U^{NEG}(Y_s^-)\|^2 \tag{4}$$

Further, we define L_{offset} as the distance between $distance_V$ and $distance_{\mathbb{D}}$:

$$L_{offset} = \|distance_V(v_j, V_s^-) - distance_{\mathbb{D}}(d_j, \mathbb{D}_s^-)\| \tag{5}$$

That is, $distance_V(v_j, V_s^-)$ gives the distance between the embedding from the KG for the positive sample and the

center of the negative ones, while $distance_{\mathbb{D}}(d_j, \mathbb{D}_s^-)$ gives the distance between the embedding from the document for the positive sample and the center of the negative ones. In standard approaches, these distances may differ greatly; in such cases, L_{offset} will be large. In contrast, DOUBLER aims to minimize L_{offset}; hence, the embedding distance between v_j and d_j to the respective negative embedded samples V_s^- and $\mathbb{D}_s^-$ will be minimized. As a result, the model explicitly accounts for similarity in the KG when learning embeddings for the documents and vice versa. That is, the model explicitly unifies the embeddings of the KG structure and the associated documents. This distinguishes DOUBLER from other methods.

It is important to note that the model adjusts the learned embeddings of the entities to find a balance between the evidence in the KG and the documents. Figure 2 (left) illustrates an example how this can affect the embedding space. The figure shows the embeddings learned by standard approaches: Gene 1017 is moderately close in the embedding space to the other Asthma-associated proteins due to the similar documents; however, Gene 1017 is not close to the others in the KG, so it is not clear if it should be associated with Asthma. On the right side of the figure, though, we see the impact of using L_{offset}. Due to this unified learning, the document embeddings inform the KG embeddings. Consequently, in this example, it is now clear that Gene 1017 should be associated with Asthma.

3.3 Learning accurate relation predictions

In addition to L_{offset}, the prediction accuracy of the model for predicting relationships is also incorporated into the learning. In particular, it uses the learned weights of the embedding layers (Emb_V, $Emb_{\mathbb{D}}$, and Emb^R) to predict the relationships known to be in the KG. More specifically, recall that we aim to predict the true tail of a triple when the head and relationship type are fixed. That is, we are given h ("Cardiomyopathy" in the example) and r_k ("`associated_with`"), and we wish to predict a correct t for the triple (h, r, t). We refer to the node and document associated with the head as v_{head} and d_{head}, respectively.

In contrast to L_{offset}, the prediction accuracy loss L_{pred} does not average the embeddings of the negative samples (from V_s^- and $\mathbb{D}_s^-$) but rather compares all of them to the correct tail (v_j and d_j) to measure the predictive performance. In particular, the embeddings of the fixed head and relationship are combined (multiplied element-wise, "·") to group the nodes in the embedding space by the relation type.

$$Group_U(x_{head}, r_k) = Emb_U(x_{head}) \cdot Emb^R(r_k) \quad (6)$$

Finally, the score of each candidate tail (that is, j as well as each negative sample) is calculated as the dot product of its embedding with $Group_U(x_{head}, r_k)$.

$$Predict_U(x_{head}, r_k, x_j) = Group_U(x_{head}, r_k) * Emb_U(x_j) \quad (7)$$

Each score is a measure of how likely the respective entity correctly completes the triple $(h, r, ?)$. In the optimal case, the score of the negative samples should be 0 while the true tail should be 1. This step is performed separately for both modalities, and the resulting vectors are combined (added element-wise).

$$Predict(head, r_k, j) = Predict_V(v_{head}, r_k, v_j) + Predict_{\mathbb{D}}(d_{head}, r_k, d_j) \quad (8)$$

We compute this score for all negative samples and the true tail. Since we consider two modalities, the combined score for the true tail should be 2, while the combined score for the negative samples should still be 0. The resulting scores are considered to compute a binary cross entropy loss.

$$L_{pred} = BCE\left(\begin{bmatrix} Predict(head, r_k, j) \\ Predict(head, r_k, y_1^-) \\ \vdots \\ Predict(head, r_k, y_m^-) \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right) \quad (9)$$

where y_i^- is the ith negative sample and BCE is the binary cross-entropy.

Finally, both loss values are combined to measure the performance of the model.

$$L = \alpha \cdot L_{pred} + (1 - \alpha) \cdot L_{offset} \quad (10)$$

where $\alpha \in [0, 1]$ is a hyperparameter which controls the tradeoff between accuracy and the unified embeddings. We set $\alpha = 0.5$ in our experiments. Standard backpropagation algorithms are then used to update the parameters of Emb_V, Emb^R and $Emb_{\mathbb{D}}$ so that the predictions become more accurate while still respecting the unification of the KG and document embeddings; that is, the parameters are updated so that L is minimized.

As an example, we refer back to Figure 2. By taking into account the graph structure and the biomedical documents and unifying their embeddings, DOUBLER recognizes that "Gene 1017" (P3) is also associated with Asthma. As a result, we identified a novel relation ($e^{d_aw_p}$) which can be investigated in more detail.

3.4 Ranking novel relationships

After training the model, the parameters of the embedding functions Emb_V, Emb^R and $Emb_{\mathbb{D}}$ are set such that true triples result in a high $Predict(\cdot)$ value, while incorrect triples result in low values. That is, the trained model can answer queries of the following forms: $(h, r, ?)$, $(h, ?, t)$, and $(?, r, t)$. Specifically, when given two of the elements, $Predict(\cdot)$ is calculated for all possible values of the missing element of the triple. These scores are used to give a ranking from the most-likely (highest-value) to least-likely (lowest) completions of the triple. For cases in which correct completions of the query are known, such rankings can be evaluated using standard information retrieval metrics.

4 Results and Discussion

In the following, we describe the conducted experiments and present our results. We compare the performance of DOUBLER with state-of-the-art techniques by predicting disease–protein associations for each disease in our dataset. We present the overall performance but also investigate for which diseases in particular our approach performs better or worse than other techniques. We also characterize the types of diseases for which our method performs well.

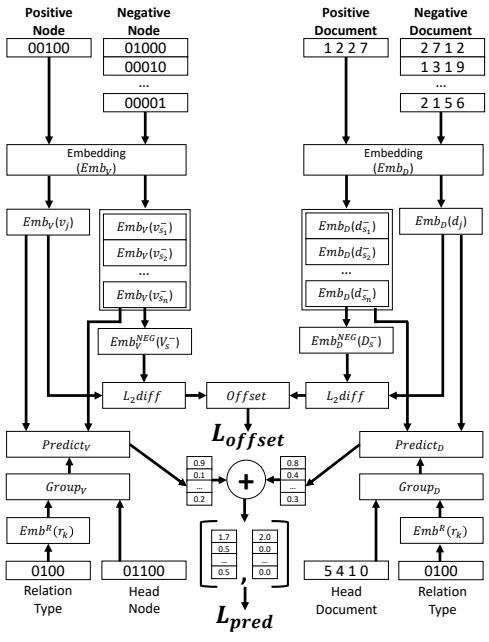

Fig. 3: Detailed neural network architecture.

4.1 Experimental design

Dataset. We created a knowledge graph (KG) as described in Section 2.1. We retained only those diseases with at least 10 associated genes. Table 1 shows the characteristics of the knowledge graph.

Proteins (P)	Diseases (D)	D–D	P–D	P–P
7 442	2 264	1 501	66 771	44 669

Table 1. Characteristics of the knowledge graph

Methods. Our experiments include evaluation of DOUBLER, as well as five other baselines and state-of-the-art approaches for disease–protein association prediction, including Neural Networks (Grover and Leskovec, 2016), DIAMOnD (Ghiassian *et al.*, 2015), Neighborhood (Navlakha and Kingsford, 2010), RandomWalk (Zhou and Skolnick, 2016), and DistMult (Yang *et al.*, 2014). For DOUBLER and DistMult, we train a single model for all diseases; the remaining methods use disease-specific models.

Evaluation metrics. For evaluation, we split the genes associated with each disease into training (80%), validation (10%), and testing (10%) sets. We adopt the commonly used *Recall@100* (*R@100*) metric (Agrawal *et al.*, 2018), evaluated independently for each disease. We further performed 10-fold cross-validation to ensure robustness of the results.

Hyperparameter optimization. We did not optimize hyperparameters for DOUBLER. We set them using standard values for similar methods. Specifically, we used a batch size of 1024, generated 100 negative samples for one training sample, and used an embedding dimension of 300 (nodes) or 64 (documents), respectively. The hyperparameters of the other methods were set according to what was reported in the respective papers.

4.2 Overall performance

Figure 4 compares the performance of our method to the others. In general, the results illustrate that it is quite challenging to predict valid protein-disease relations; the best-performing methods achieve a mean *R@100* of about 0.4. Further, all of the methods exhibit high variance in performance for different diseases. Nevertheless, DOUBLER outperforms the other methods, demonstrating that incorporating the documents is beneficial.

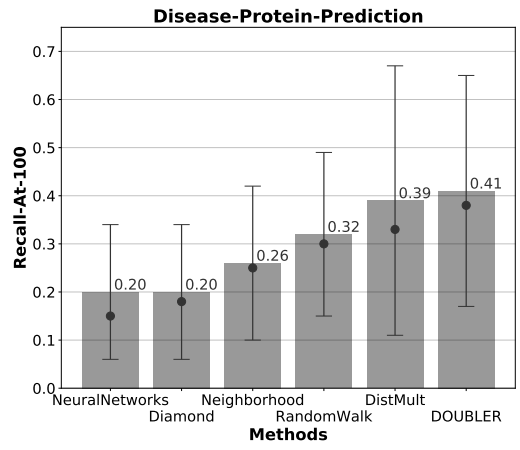

Fig. 4: Performance of all methods for predicting disease–protein associations

4.3 Convergence behavior

The best performing methods, DOUBLER and DistMult, are both sophisticated neural network-based knowledge graph embedding approaches which can be prone to overfitting. To ensure the stability of the approaches, we monitored their performance on a validation set throughout training. As Figure 5 shows, the *R@100* is quite stable across several different runs with different dataset splits. The figure also highlights that both methods reach their peak performance within a short time, and that DOUBLER continuously performs equal or better than *DistMult*.

4.4 Disease comparison

Comparing the performance of these two methods for each disease individually (Figure 6) shows that, on the one hand, DOUBLER performs similarly or better than DistMult in about 70% of all considered diseases. On the other hand, there are 61 diseases where DistMult performs better.

Disease and document characteristics

One of the main differences between DOUBLER and DistMult is the use of text documents during learning. In order to better understand the characteristics of the diseases where our approach performs better or worse than DistMult, we extracted those where the gap in *R@100* is $> 20\%$ (see Figure 7).

The related protein and document sets show some interesting differences. First, the diseases where DOUBLER performs better tend to have more associated proteins (see Figure 8);

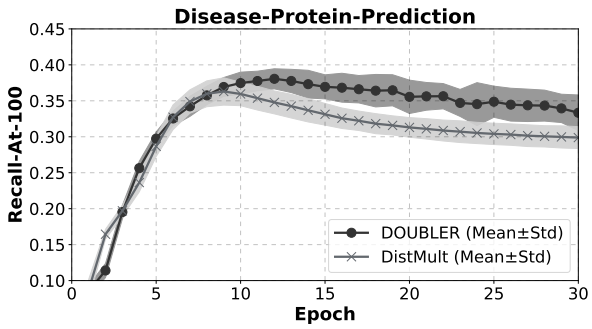

Fig. 5: Convergence behavior of DistMult and DOUBLER. The model was evaluated on a held-out validation set. The standard deviation is calculated based on the 10 cross-validation results.

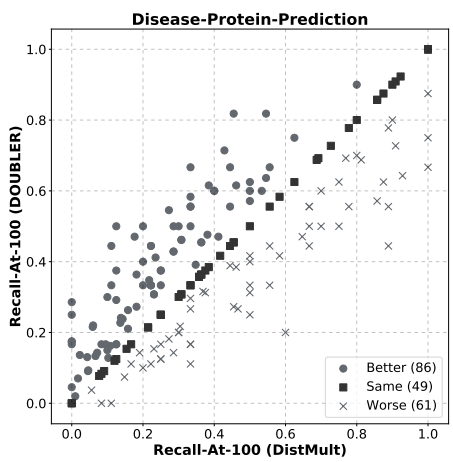

Fig. 6: Disease–protein link prediction performance: DistMult vs. DOUBLER. Each point corresponds to a single disease. Circles indicate diseases for which DOUBLER was better, crosses indicate diseases for which DistMult was better, and squares indicate diseases for which performance was equal.

consequently, more documents (and, by implication, more training data) are available. Further, Figure 9 shows that the documents related with the diseases that are better recognized by DOUBLER have more unique terms per protein. This suggests that those documents describe the proteins in more detail. Since DOUBLER incorporates the documents into the learning, it leverages these detailed descriptions to train better models.

We next analyzed the individual terms within the documents for which the performance of DistMult or DOUBLER was different. We first considered the inverse document frequency (IDF) of terms in the documents. In general, a high IDF shows that a term occurs only in a few documents, while a low IDF indicates that a term appears in many documents and may not be very informative. Indeed, our analysis (Figure 10) shows that DOUBLER outperforms DistMult when the document terms have a high IDF and are informative about the disease. Finally, we investigated the usage of each term within the documents for each disease (Figure 11). As expected, DOUBLER outperforms

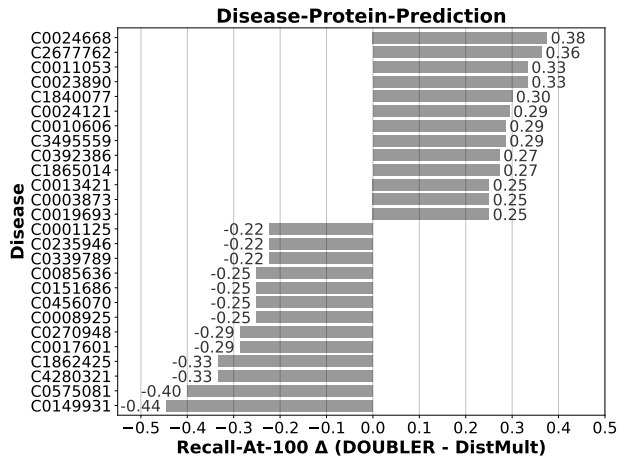

Fig. 7: Diseases with the biggest differences in performance for DistMult compared to DOUBLER. Positive values indicate diseases for which DOUBLER was better, while negative values indicate that DistMult was better.

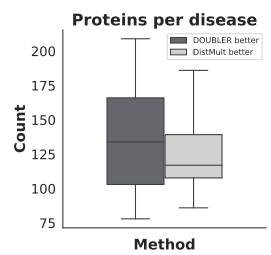

Fig. 8: Number of proteins per disease

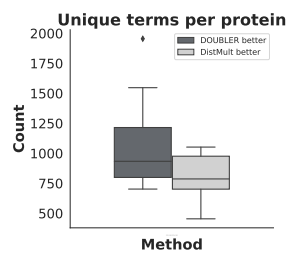

Fig. 9: Unique terms per protein

DistMult for diseases in which terms are more frequently used in multiple documents for the disease.

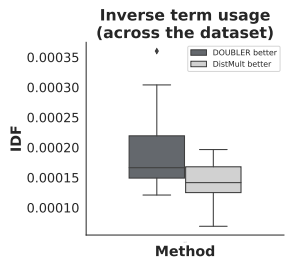

Fig. 10: Inverse term usage (across the dataset)

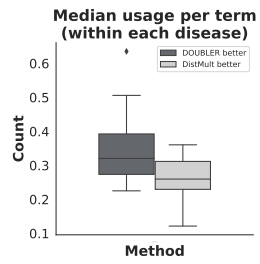

Fig. 11: Median usage per term (within each disease)

In summary, these results paint a consistent picture in which DOUBLER outperforms DistMult and other state-of-the-art approaches for complex diseases which involve many genes (Figure 8), and also where the individual genes within the disease are consistently described (Figures 9–11). While these results are not surprising considering that a key difference of DOUBLER compared to others is its use of documents, these

quantitative and qualitative experiments confirm this expected behavior.

5 Conclusions and Future Work

In this work, we have presented a general framework for learning representations of biological entities which are consistent among both data attributes and the network structure. In particular, we incorporate a Euclidean distance offset between embeddings based on the knowledge graph structure and the documents associated with entities into the loss function for training a neural network; thus, the model explicitly unifies the embeddings of the KG structure and the associated documents. This distinguishes our approach from other proposed methods. We demonstrated that this approach outperforms competing methods, and more detailed analysis showed it is particularly beneficial when the data attributes are informative. Future work will evaluate the efficacy of our approach for more comprehensive knowledge graphs, such as the recent OpenBioLink (Breit *et al.*, 2020) dataset. Further, it is well known (Kadlec *et al.*, 2017) that hyperparameter tuning can lead to significant performance gains for many methods; thus, future work will also incorporate hyperparameter optimization strategies.

References

Agrawal, M., Zitnik, M., and Leskovec, J. (2018). Large-scale analysis of disease pathways in the human interactome. In *Proceedings of the Pacific Symposium on Biocomputing*.

Asif, M., Martiniano, H. F. M. C. M., Vicente, A. M., and Couto, F. M. (2018). Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology. *PLOS ONE*, **13**(12), e0208626.

Ata, S. K., Ou-Yang, L., Fang, Y., Kwoh, C.-K., Wu, M., and Li, X.-L. (2018). Integrating node embeddings and biological annotations for genes to predict disease–gene associations. *BMC Systems Biology*, **12**(9).

Breit, A., Agibetov, A., and Samwald, M. (2020). OpenBioLink: A resource and benchmarking framework for large-scale biomedical link prediction. *Bioinformatics*, **36**(13), 4097–4098.

Caldera, M., Buphamalai, P., Müller, F., and Menche, J. (2017). Interactome-based approaches to human disease. *Current Opinion in Systems Biology*, **3**, 88–94.

Gene Ontology Consortium (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, **32**(suppl_1), D258–D261.

Ghiassian, S. D., Menche, J., and Barabási, A.-L. (2015). A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLOS Computational Biology*, **11**(4), e1004120.

Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. (2011). PREDICT: A method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, **7**(1).

Grover, A. and Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the* 22nd *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Härtner, F., Andrade-Navarro, M. A., and Alanis-Lobato, G. (2018). Geometric characterisation of disease modules. *Applied Network Science*, **3**(1), 1–17.

Kadlec, R., Bajgar, O., and Kleindienst, J. (2017). Knowledge base completion: Baselines strike back. In *Proceedings of the* 2nd *Workshop on Representation Learning for NLP*.

Köhler, S., Doelken, S. C., Mungall, C. J., Bauer, S., Firth, H. V., Bailleul-Forestier, I., Black, G. C., Brown, D. L., Brudno, M., Campbell, J., *et al.* (2014). The Human Phenotype Ontology project: Linking molecular biology and disease through phenotype data. *Nucleic Acids Research*, **42**(D1), D966–D974.

Kotnis, B. and Nastase, V. (2018). Analysis of the impact of negative sampling on link prediction in knowledge graphs. In *Proceedings of the* 1st *Workshop on Knowledge Base Construction, Reasoning and Mining*.

Lo Surdo, P., Calderone, A., Iannuccelli, M., Licata, L., Peluso, D., Castagnoli, L., Cesareni, G., and Perfetto, L. (2017). DISNOR: A disease network open resource. *Nucleic Acids Research*, **46**(D1), D527–D534.

Luo, P., Tian, L.-P., Ruan, J., and Wu, F.-X. (2017). Disease gene prediction by integrating PPI networks, clinical RNA-Seq data and OMIM data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **16**(1), 222–232.

Madeddu, L., Stilo, G., and Velardi, P. (2019). Network-based methods for disease–gene prediction. *arXiv.org*. 1902.10117.

Malone, B., García-Durán, A., and Niepert, M. (2018). Knowledge graph completion to predict polypharmacy side effects. In *Proceedings of the* 13th *International Conference on Data Integration in the Life Sciences*.

Mohamed, S. K., Nováček, V., and Nounu, A. (2020). Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics*, **36**(2), 603–610.

Navlakha, S. and Kingsford, C. (2010). The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, **26**(8), 1057–1063.

Nováček, V. and Mohamed, S. K. (2020). Predicting polypharmacy side-effects using knowledge graph embeddings. In *Proceedings of the AMIA Joint Summits on Translational Science*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the* 54th *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sharma, A., Menche, J., Huang, C. C., Ort, T., Zhou, X., Kitsak, M., Sahni, N., Thibault, D., Voung, L., Guo, F., *et al.* (2015). A disease module in the interactome explains disease heterogeneity, drug response and captures novel pathways and genes in asthma. *Human Molecular Genetics*, **24**(11), 3005–3020.

Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N. T., Morris, J. H., Bork, P., *et al.* (2018). STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, **47**(D1), D607–D613.

Vlaic, S., Conrad, T., Tokarski-Schnelle, C., Gustafsson, M., Dahmen, U., Guthke, R., and Schuster, S. (2018). ModuleDiscoverer: Identification of regulatory modules in protein–protein interaction networks. *Scientific Reports*, **8**(433).

Wu, C., MacLeod, I., and Su, A. I. (2013). BioGPS and MyGene.info: Organizing online, gene-centric information. *Nucleic Acids Research*, **41**(D1), D561–D565.

Yang, B., Yih, S. W.-t., He, X., Gao, J., and Deng, L. (2014). Learning multi-relational semantics using neural-embedding models. In *NIPS 2014 Workshop on Learning Semantics*.

Zhou, H. and Skolnick, J. (2016). A knowledge-based approach for predicting gene–disease associations. *Bioinformatics*, **32**(18), 2831–2838.

Zhou, J. and Fu, B.-q. (2018). The research on gene-disease association based on text-mining of PubMed. *BMC Bioinformatics*, **19**(37).