Multimodal Learning on Graphs for Disease Relation Extraction

Yucong Lin\textsuperscript{1,2}, Keming Lu\textsuperscript{3}, Sheng Yu\textsuperscript{4,5}, Tianxi Cai\textsuperscript{6,7}, and Marinka Zitnik\textsuperscript{7,8,9}

\textsuperscript{1}Institute of Engineering Medicine, Beijing Institute of Technology, Beijing, China
\textsuperscript{2}School of Optics and Photonics, Beijing Institute of Technology, Beijing, China
\textsuperscript{3}Viterbi School of Engineering, University of Southern California
\textsuperscript{4}Center for Statistical Science, Tsinghua University, Beijing, China
\textsuperscript{5}Department of Industrial Engineering, Tsinghua University, Beijing, China
\textsuperscript{6}Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA 02115, USA
\textsuperscript{7}Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, USA
\textsuperscript{8}Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA
\textsuperscript{9}Harvard Data Science Initiative, Cambridge, MA 02138, USA

Abstract

Objective: Disease knowledge graphs are a way to connect, organize, and access disparate information about diseases with numerous benefits for artificial intelligence (AI). To create knowledge graphs, it is necessary to extract knowledge from multimodal datasets in the form of relationships between disease concepts and normalize both concepts and relationship types.

Methods: We introduce REMAP, a multimodal approach for disease relation extraction and classification. The REMAP machine learning approach jointly embeds a partial, incomplete knowledge graph and a medical language dataset into a compact latent vector space, followed by aligning the multimodal embeddings for optimal disease relation extraction.

Results: We apply REMAP approach to a disease knowledge graph with 96,913 relations and a text dataset of 1.24 million sentences. On a dataset annotated by human experts, REMAP improves text-based disease relation extraction by 10.0\% (accuracy) and 17.2\% (F1-score) by fusing disease knowledge graphs with text information. Further, REMAP leverages text information to recommend new relationships in the knowledge graph, outperforming graph-based methods by 8.4\% (accuracy) and 10.4\% (F1-score).

Discussion: Systematized knowledge is becoming the backbone of AI, creating opportunities for disease knowledge graphs and text-based disease relation extraction.
to inject semantics into AI and fully integrate it into machine learning algorithms. While prior semantic knowledge can assist in extracting disease relationships from text, existing methods can not fully leverage multimodal datasets.

**Conclusion:** REMAP is a multimodal approach for extracting and classifying disease relationships by fusing structured knowledge and text information. REMAP provides a flexible neural architecture to easily find, access, and validate AI-driven relationships between disease concepts.

**Key words:** Disease relation extraction, Medical relation extraction, Multimodal learning, Knowledge graphs, Graph neural networks, Language neural models

Word count (excluding title page, abstract, references, figures and tables): 4,234

1 **BACKGROUND AND SIGNIFICANCE**

Disease knowledge graphs are a way to connect, organize, and access disparate data and information resources about diseases with numerous benefits for artificial intelligence (AI). Systematized knowledge can be injected into AI tools to imitate human experts’ reasoning so that AI-driven hypotheses can be easily found, accessed, and validated. For example, disease knowledge graphs (KGs) power AI applications, such as intelligent diagnosis and electronic health record (EHR) retrieval. However, creating high-quality knowledge graphs requires extracting relationships between diseases from disparate information sources, such as free text in the EHRs and semantic knowledge representation from literature.

Traditionally, KGs were constructed via manual efforts, requiring humans to input every fact [1]. In contrast, rule-based [2] and semi-automated [3, 4] construction, while scalable, can suffer from poor accuracy and recall rate. Recent methods create KGs by extracting relationships from literature and building considerably more accurate KGs that comprehensively cover a domain of interest. These methods leverage pre-trained language models [5, 6] and have advanced biomedical knowledge graphs [7, 8, 9, 10, 11]. Another approach for extracting relations uses knowledge graph embeddings (KGE), which directly predict new relations in partial, incomplete knowledge graphs. KGE methods learn how to represent every entity (i.e., node) and relation (i.e., edge) in a graph as a distinct point in a low-dimensional vector space.
(i.e., embedding) so that performing algebraic operations in this learned space reflects the topology of the graph [12]. Embeddings produced by KGE methods can be remarkably powerful for downstream AI applications [13, 14, 15, 16, 17, 18]. Widely used KGE methods include translation models [19, 20, 16, 21], bilinear models [22, 23, 24, 25], and graph neural networks (GNNs) [26, 27, 28, 29]. These methods leverage embeddings to predict new relations, thereby completing and growing an existing KG. However, extracting relations from a single data type may suffer from bias, noise, and inherent incompleteness. For example, in language-based methods, the training dataset is collected using distant supervision [30], which creates noisy sentences that can mislead relation extraction. Further, graph-based methods can suffer from the out-of-dictionary problem, limiting the ability to predict relations involving entities previously not in the KG [19, 31].

Nevertheless, language-based and graph-based methods both have advantages. For example, language-based methods can reason over large datasets created using distant supervision, and graph-based methods can operate on noisy and incomplete knowledge graphs, providing robust predictions. An emerging strategy to advance relation extraction thus leverages multiple data types simultaneously [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 20, 42, 43, 44, 45] with multimodal learning, outperforming rule-based [2] and semi-automated [3, 4] methods. However, existing approaches are limited in two ways, which we outline next.

First, KGs provide only positive samples (i.e., examples of true disease-disease relationships) while existing methods also require negative samples that, ideally, are disease pairs that resemble positive samples but are not valid relationships. Methods for positive-unlabeled [46] and contrastive [47, 48] learning can be trained in such scenarios using random disease pairs from the dataset as negative proxy samples and ensuring a low false-positive rate. However, these methods may not generalize well in real-world applications because random negative samples are not necessarily realistic and fail to represent the boundary cases. To improve the quality of negative sampling in disease relation extraction, we introduce an EHR-based negative sampling strategy. With the strategy, our approach generates negative samples using disease pairs that rarely appear together in EHRs, thus having realistic negative samples to enable broad generalization of the approach.

Second, it is not uncommon for certain diseases to appear either in graph or language datasets
but not in both data types. For example, in Lin et al. [9], over 60% disease pairs in the knowledge graph had no corresponding text information; there were also cases with text but no graph information. It is thus essential that multimodal approaches can make predictions when only one data type is available. Unfortunately, existing multimodal approaches with such capability [49, 50, 51, 52] do not focus on language and graphs. Further, missing data imputation techniques use adversarial learning to impute missing values [53, 54], but the imputed values can introduce unwanted data bias. To address this issue, we propose a multimodal approach with a decoupled model structure where language-based and graph-based modules interact only through shared parameters and a cross-modal loss function. This method ensures our model can take advantage of both language and graph inputs and extract disease relations based on either single or multimodal inputs.

**Present work.** We introduce REMAP (Relation Extraction with Multimodal Alignment Penalty)\(^1\), a multimodal approach for extracting and classifying disease-disease relations (Figure 1). REMAP is a flexible multimodal algorithm that jointly learns over text and graphs with a unique capability to make predictions even when a disease concept exists in only one data type. To this end, REMAP specifies graph-based and text-based deep transformation functions that embed each data type separately and optimize unimodal embedding spaces such that they capture the topology of a disease KG or the text semantics of disease concepts. Finally, to achieve data fusion, REMAP aligns unimodal embedding spaces through a novel alignment penalty loss using shared disease concepts as anchors. This way, REMAP can effectively model data type-specific distribution and diverse representations while also aligning embeddings of distinct data types. Further, REMAP can be jointly trained on both graph and text data types but evaluated and implemented on either of the two modalities alone. In summary, the main contributions of this study are:

- We develop REMAP, a flexible multimodal approach for extracting and classifying disease-disease relations. REMAP fuses knowledge graph embeddings with deep language models and can flexibly accommodate missing data types, which is a necessary capability to facilitate REMAP’s validation and transition into biomedical implementation.

---

\(^1\)Python implementation of REMAP is available on Github at [https://github.com/Lukeming-tsinghua/REMOD](https://github.com/Lukeming-tsinghua/REMOD). Our dataset of domain-expert annotations is at [https://doi.org/10.6084/m9.figshare.17776865](https://doi.org/10.6084/m9.figshare.17776865).
• We rigorously evaluate REMAP for extraction and classification of disease-disease relations. To this end, we create a training dataset using distant supervision and a high-quality test dataset of gold-standard annotations provided by three domain experts, all medical doctors. Our evaluations show that REMAP achieves 88.6% micro-accuracy and 81.8% micro-F1 score on the human annotated dataset, outperforming text-based methods by 10 and 17.2 percentage points, respectively. Further, REMAP achieves the best performance, 89.8% micro-accuracy and 84.1% micro-F1 score, surpassing graph-based methods by 8.4 and 10.4 percentage points, respectively.

2 METHODS

We next detail the REMAP approach and illustrate it in detail for the task of disease relation extraction and classification (Figure 1). We first describe the notation, proceed with an overview of language and knowledge graph models, and outline the multimodal learning strategy to inject knowledge into extraction tasks.

2.1 Preliminaries

Notation. The input to REMAP is a combined dataset $D$ of text and graph information. This dataset consists of text information $D^T = \{B_i\}_{i=1}^{M_0} = D^T_L \cup D^T_U = \{B_i\}_{i=1}^M \cup \{B_i\}_{j=M+1}^{M_0}$ given as $M_0$ bags of sentences and graph information $D^G = \{(s_i, r_i, o_i)\}_{i=1}^M \cup \{(s_i, r_i, o_i)\}_{i=M+1}^N$ given as $N$ triplets $(s_i, r_i, o_i)$ encoding the relationship between $s_i$ and $o_i$ as $r_i$. For example, $s_i$ = “hypogonadism”, $o_i$ = “Goldberg-Maxwell syndrome” and $r_i$ = “Differential diagnosis” would indicate the process to differentiate between hypogonadism versus Goldberg-Maxwell syndrome with similar symptoms that could possibly account for illness in a patient. We assume that $M$ bags of sentences in $D^T$ overlap with the triplets from existing KG such that each sentence of the $i$th sentence bag contain $(s_i, r_i)$. The remaining $N - M$ triplets in $D^G$ cannot be mapped to sentences in $D^T$ and $M_0 - M$ sentences contain entity pairs that do not belong to existing KG. We represent the $i$-th sentence bag as $B_i = \{(t_{ij}, I^T_{ij}, I^O_{ij})\}_{j=1}^{l_i}$, where $l_i$ is the number of sentences in bag $B_i$, $t_{ij}$ is the tokenized sequence of $j$-th sentence in $B_i$. Here, the tokenized sequence is the combination of sentence containing mentions of subject and object entities, entity markers,
title of the document in which the the sentence appears, and the article structure. Marker tokens are added to the head and tail position of each entity mentions to denote entity type information. Last, \( I^s_{ij} \) and \( I^o_{ij} \) are start indices of entity markers for subject and object entities, respectively.

**Heterogeneous graph attention network.** Heterogeneous graph attention network (HAN) [28] is a graph neural network for embedding KGs by leveraging meta paths. A meta path is a sequence of node types and relation types [55]. For example, in a disease KG, “Disease” → “May Cause” → “Disease” → “Differential Diagnosis” → “Disease” is a meta path. Node \( u_i \) is connected to node \( u_j \) via a meta path \( \Phi \) if \( u_i \) and \( u_j \) are the head and tail nodes, respectively, of this meta path. Each node \( u_i \) has an initial node embedding \( h^\text{init}_i \) and belongs to a type \( \phi_i \), e.g., \( \phi_i = \text{“disease concept”} \).

Graph attention network specified a parameterized deep transformation function \( f_{\text{HAN}} \) that maps nodes to condensed data summaries, i.e., embeddings, in a node-type specific manner as:

\[
h'_i = f_{\text{HAN}}(h^\text{init}_i).
\]

We denote all nodes adjacent to \( u_i \) via a meta-path \( \Phi \) as \( u_j \in N^\Phi_i \) and node-level attention mechanism provides information on how strongly \( f_{\text{HAN}} \) attends to \( u_i \)’s each adjacent node \( u_j \) when generating the embedding for \( u_i \). In particular, the importance of \( u_j \) for \( u_i \) in meta path \( \Phi \) is defined as:

\[
a^\Phi_{ij} = \frac{\exp(\sigma(a^T_{ij} \cdot [h'_i || h'_j]))}{\sum_{k \in N^\Phi_i} \exp(\sigma(a^T_k \cdot [h'_i || h'_k]))},
\]

where \( \sigma \) is the sigmoid activation, \( || \) indicates concatenation, and \( a^\Phi \) is a trainable vector. To promote stable attention, HAN uses the above specified node-level attention to produce the final node embedding for node \( u_i \):

\[
z^\Phi_i = ||_{k=1}^K \sigma(\sum_{j \in N^\Phi_i} a^\Phi_{ij} \cdot h'_j),
\]

Given user-defined meta paths \( \Phi_1, \ldots, \Phi_P \), HAN uses the above specified node-level attention to produce node embeddings \( Z_{\Phi_1}, \ldots, Z_{\Phi_P} \). Finally, HAN uses semantic-level attention to combine meta path-specific node embeddings as:

\[
\beta_{\Phi_p} = \frac{\exp(\frac{1}{|E|} \sum_{i \in E} q^T \cdot \tanh(W \cdot z^\Phi_p + b))}{\sum_{p=1}^P \exp(\frac{1}{|E|} \sum_{i \in E} q^T \cdot \tanh(W \cdot z^\Phi_i + b))},
\]
\[
Z = \sum_{p=1}^{P} \beta_{\Phi_p} \cdot Z_{\Phi_p},
\]
where \( \beta_{\Phi_p} \) represents the importance of meta path \( \Phi_p \) towards final node embeddings \( Z \), and \( q^T, W, \) and \( b \) are trainable parameters. The final outputs are node embeddings \( Z \), representing compact vector summaries of knowledge associated with each node in the KG.

**Translation- and tensor-based embeddings.** TransE [19] and TuckER [25] decode an optimized set of embeddings into the probability estimate of relationship \( r_i \) existing between entities \( s_i \) and \( o_i \). This is achieved by a scoring function (SF) that either translates the embeddings in TransE as:
\[
SF_{Tr}(h_{s_i}, h_{o_i}, h_r) = \sigma(||h_{s_i} + h_r - h_{o_i}||_2^2)
\]
or factorizes the embeddings in TuckER as:
\[
SF_{Tu}(h_{s_i}, h_{o_i}, h_r, W) = \sigma(W \times_1 h_{s_i} \times_2 h_r \times_3 h_{o_i}),
\]
where \( h_{s_i}, h_{o_i}, \) and \( h_r \) represent entity and relation embeddings, \( \sigma \) denotes the sigmoid function, \( W \in \mathbb{R}^{d_{s_i} \times d_r \times d_{o_i}} \) is a trainable tensor, and \( \times_d, d = 1, 2, 3, \) indicates tensor multiplication along dimension \( d \).

### 2.2 Language and knowledge graph encoders

**Embedding disease-associated sentences.** We start by tokenizing entities in sentences and proceed with an overview of the text encoder. Entity tokens identify the position and type of entities in a sentence [7, 8]. Specifically, tokens \(<S\text{-type}>\) and \(<S\text{-type}/>\), and \(<O\text{-type}>\) and \(<O\text{-type}/>\) are used to denote the start and end of subject (S) and object (O) entities, respectively. The entity marker tokens are type-related, meaning that entities of different types (e.g., disease concepts, medications) get different tokens. This procedure produces bags of tokenized sentences \( D_T \) that we encode into entity embeddings using a text encoder. We use SciBERT encoder [6] with the SciVocab wordpiece vocabulary, which is a BERT language model optimized for scientific text with improved efficiency in biomedical domains than BioBERT or BERT model alone [8]. Tokenized sequences in a sentence bag \( B_i \) are fed into the language model to produce a set of sequence outputs \( \mathbb{H}_i = \{H_i^1, \ldots, H_i^{l_i}\}, \)
\[
\mathbb{H}_i^{[m]} = \text{SciBERT}(B_i; \mathbb{H}_i^{[m-1]}),
\]
where \( \mathbb{H}_i = [H_{i1}, \ldots, H_{il_i}], \) \( l_i \) is the number of sentences in \( B_i \), \( H_i \in \mathbb{R}^{d_{s_i} \times d_{hs}}. \) We then aggregate representations of the subject entity \( s_i \) across all sentences in bag \( B_i \) as: \( H_{s_i} = \| \sum_{k \in \mathbb{T}_i^j} H_{mk} \|_m \).
and use self-attention to obtain the final text-based embedding $h^T_{s_i}$ for subject entity as:

$$h^T_{s_i} = H_{s_i} \cdot \text{softmax}(\omega \cdot \text{tanh}(H_{s_i})), \quad (6)$$

where $h^T_{s_i} \in \mathbb{R}^{d_{hs}}$ is the embedding of $s_i$ and $\omega$ is a trainable vector. Embeddings of object entities (i.e., $h^T_{o_i}$ for object entity $o_i$) are generated analogously by the text encoder. Self-attention is needed because some sentences in each bag may not capture any relationship between entities, and attention allows the model to ignore such uninformative sentences when generating embeddings.

**Embedding a disease knowledge graph.** We use a heterogeneous graph attention encoder [28] to map nodes in the disease KG $D^G$ into embeddings. The encoder generates embeddings for every subject entity $h^G_{s_i}$ and object entity $h^G_{o_i}$ that appear as nodes in the KG as:

$$h^G_{s_i} = f_{\text{HAN}}(D^G, H^{\text{init}}, s_i), \quad h^G_{o_i} = f_{\text{HAN}}(D^G, H^{\text{init}}, o_i), \quad (7)$$

where $f_{\text{HAN}}$ transformation is given in Eqs. (1)-(4) and $H^{\text{init}}$ denotes initial embeddings.

**Scoring disease-disease relationships.** Taking text-based embeddings, $h^T_{s_i}, h^T_{o_i}$, and graph-based embeddings, $h^G_{s_i}, h^G_{o_i}$, for every subject and object occurring in either language or graph dataset, we use a scoring function to estimate disease-disease relationships formulated as candidate triplets $(s_i, r_k, o_i)$:

$$p^T(r_{ik} = 1|s_i, o_i) = \text{SF}(h^T_{s_i}, h^T_{o_i}, h^T_r), \quad k = 1, 2, \ldots, K, \quad (8)$$

$$p^G(r_{ik} = 1|s_i, o_i) = \text{SF}(h^G_{s_i}, h^G_{o_i}, h^G_r), \quad k = 1, 2, \ldots, K, \quad (9)$$

where SF is the scoring function and $K$ denotes the number of relation types. We consider three scoring functions, including the linear scoring function: $\text{SF}_{\text{Li}}(h_{s_i}, h_{o_i}, h_{r_k}) = \sigma(W_k(h_{s_i} + h_{o_i}) + b_k)$, the TransE scoring function: $\text{SF}_{\text{Tr}}(h_{s_i}, h_{o_i}, h_{r_k}) = \sigma(||h_{s_i} + h_{r_k} - h_{o_i}||_2^2)$, and the TuckER scoring function: $\text{SF}_{\text{Tu}}(h_{s_i}, h_{o_i}, h_{r_k}, W) = \sigma(W \times_1 h_{s_i} \times_2 h_{r_k} \times_3 h_{o_i})$. In the TuckER function, we use a separate kernel $W^T$ and $W^G$ for text and graph data types. In the TransE function, $h_{r_k}$ is the embedding of relation $r_k$ shared between modalities (Section 2.1).
2.3 Co-training language and graph encoders

We proceed describing the procedure for co-training text and graph encoders. From last section, we obtain relationship estimates based on evidence provided by text information, \( p^T(r_{ik} = 1|s_i, o_i) \), and graph information, \( p^G(r_{ik} = 1|s_i, o_i) \), for every triplet \((s_i, r_i, o_i)\). We use the binary cross entropy to optimize those estimates in each data type as:

\[
L^T = \sum_{i=1}^{M} \sum_{k=1}^{K} r_{ik} \log(p^T(r_{ik} = 1|s_i, o_i)) + (1 - r_{ik}) \log(1 - p^T(r_{ik} = 1|s_i, o_i)),
\]

(10)

\[
L^G = \sum_{i=1}^{M} \sum_{k=1}^{K} r_{ik} \log(p^G(r_{ik} = 1|s_i, o_i)) + (1 - r_{ik}) \log(1 - p^G(r_{ik} = 1|s_i, o_i)),
\]

(11)

and can combine text-based loss \( L^T \) and graph-based loss \( L^G \) as: \( L_{\text{REMAP}} = L^T + L^G \). This loss function is motivated by the principle of knowledge distillation [56] to enhance multimodal interaction and improve classification performance. Using probabilities \( p^T(r_{ik} = 1|s_i, o_i) \) from the text encoder, we normalize them across \( K \) relation types to obtain a text-based distribution \( p^T(s_i, o_i) \) as:

\[
p^T(s_i, o_i) = \left\{ \frac{e^{p^T(r_{ik} = 1|s_i, o_i)}}{\sum_{m=1}^{K} e^{p^T(r_{im} = 1|s_i, o_i)}} \right\}_{k=1}^{K}.
\]

(12)

We calculate a graph-based distribution \( p^G(s_i, o_i) \) in the same manner using softmax normalization.

Specifically, we develop two REMAP variants, REMAP-M and REMAP-B, based on how text-based and graph-based losses are combined into a multimodal objective. In REMAP-M, text-based and graph-based losses are aligned by shrinking the distance between \( p^T(s_i, o_i) \) and \( p^G(s_i, o_i) \) using Kullback-Leibler (KL) divergence:

\[
D_{\text{KL}}(p^T, p^G) = \sum_{i=1}^{M} \sum_{k=1}^{K} p^G(s_i, o_i)_k \log \left( \frac{p^G(s_i, o_i)_k}{p^T(s_i, o_i)_k} \right),
\]

(13)

as a measure of misalignment as follows:

\[
L_{\text{REMAP-M}} = L_{\text{REMAP}} + \lambda_M(D_{\text{KL}}(p^T, p^G) + D_{\text{KL}}(p^G, p^T)).
\]

(14)

In contrast, REMAP-B considers the best logit on two data types, inspired by ensemble distillation [57]. Specifically, if \( r_{ik} = 1 \), the highest predicted score across data types is used as the best
score \( p^B \):

\[
p^B(r_{ik} = 1|s_i, o_i) = \begin{cases} 
  p^T(r_{ik} = 1|s_i, o_i), & p^T(r_{ik} = 1|s_i, o_i) \leq p^G(r_{ik} = 1|s_i, o_i) \text{ and } r_{ik} = 0 \\
  p^T(r_{ik} = 1|s_i, o_i), & p^T(r_{ik} = 1|s_i, o_i) \geq p^G(r_{ik} = 1|s_i, o_i) \text{ and } r_{ik} = 1 \\
  p^G(r_{ik} = 1|s_i, o_i), & p^T(r_{ik} = 1|s_i, o_i) \geq p^G(r_{ik} = 1|s_i, o_i) \text{ and } r_{ik} = 0 \\
  p^G(r_{ik} = 1|s_i, o_i), & p^T(r_{ik} = 1|s_i, o_i) \leq p^G(r_{ik} = 1|s_i, o_i) \text{ and } r_{ik} = 1, 
\end{cases}
\]

(15)

which is normalized across \( K \) relation types using a softmax function:

\[
p^B(s_i, o_i) = \left\{ e^{p^B(r_{ik} = 1|s_i, o_i)} \middle/ \sum_{k=1}^{K} e^{p^B(r_{ik} = 1|s_i, o_i)} \right\}^{K}_{k=1}.
\]

(16)

REMAP-B uses cross entropy to quantify the discrepancy between predicted scores and ground truth labels:

\[
L^B = \sum_{i=1}^{M} \sum_{k=1}^{K} r_{ik} \log(p^B(r_{ik} = 1|s_i, o_i)) + (1 - r_{ik}) \log(1 - p^B(r_{ik} = 1|s_i, o_i)),
\]

(17)

and co-trains graph and text encoders by penalizing misaligned encoders:

\[
L_{REMAP-B} = L_{REMAP} + \lambda_B[L^B + D_{KL}(p^B, p^T) + D_{KL}(p^B, p^G)].
\]

(18)

The outline of complete REMAP-B is shown in Algorithm 1.

### 3 DATA AND EXPERIMENTAL SETUP

We proceed with the description of datasets (Section 3.1), followed by the training and implementation details of REMAP approach (Section 3.2) and the outline of experimental setup (Section 3.3).

#### 3.1 Datasets

Datasets used in this study are multimodal and originate from diverse sources that we integrated and harmonized as outlined below. In particular, we compiled a large disease-disease knowledge
Algorithm 1: REMAP, multimodal learning on graphs for disease relation extraction and classification. Shown is the outline of REMAP-B (Section 2.3).

Input: Bag of sentences \( \{B_i\}_{i=1}^{n} \), Knowledge graph \( D^G = \{(s_i, r_i, o_i)\} \), Initial node embeddings \( H^{init} \), Scoring function SF, Regularization strength \( \lambda_B \), Relation type vector pretrained on text \( r^T \), Relation type vector pretrained on graph \( r^G \)

Output: Model parameters \( \Theta \) of text-based and graph-based encoders

1. Initialize model parameters \( \Theta \)
2. Initialize relation representation as \( r = \frac{r^T + r^G}{2} \)
3. for epoch=1:n_epochs do
   4. for \( i=1:n_{\text{bags}} \) do
   5. /* Encode text and calculate text logits, see Section 2.2 */
      6. \( H = \text{SciBERT}(B_i) \)
      7. \( H_{s_i} = \|_{m=1}^{l_i} \|_{k \in I^G_{s_i}} H_{mk} \)
      8. \( H_{o_i} = \|_{m=1}^{l_i} \|_{k \in I^G_{o_i}} H_{mk} \)
      9. \( h^T_i = H_{s_i} \cdot \text{softmax}(\omega \cdot \text{tanh}(H_{s_i})) \)
      10. \( h^G_i = H_{o_i} \cdot \text{softmax}(\omega \cdot \text{tanh}(H_{o_i})) \)
      11. \( p^T_i(s_i, o_i) = \text{SF}(h^T_i, h^T_i, r) \)
      12. \( L^T = \text{BinaryCrossEntropyLoss}(p^T_i(s_i, o_i), r(s_i, o_i)) \)
   5. /* Encode graph and calculate graph logits, see Section 2.2 */
   6. \( h^G_i = \text{HAN}(G, H^{init}, s_i) \)
   7. \( h^G_i = \text{HAN}(G, H^{init}, o_i) \)
   8. \( p^G_i(s_i, o_i) = \text{SF}(h^G_i, h^G_i, r) \)
   9. \( L^G = \text{BinaryCrossEntropyLoss}(p^G_i(s_i, o_i), r(s_i, o_i)) \)
   10. /* Find the best logit and calculate alignment penalty loss, see Section 2.3 */
       11. \( p^B_i(s_i, o_i) = \text{CalculateBestLogic}(p^T_i(s_i, o_i), p^G_i(s_i, o_i)) \)
       12. \( p^T_i(s_i, o_i) = \text{softmax}(p^T_i(s_i, o_i)) \)
       13. \( p^G_i(s_i, o_i) = \text{softmax}(p^G_i(s_i, o_i)) \)
       14. \( p^B_i(s_i, o_i) = \text{softmax}(p^B_i(s_i, o_i)) \)
       15. \( L^B = \sum_{i=1}^{M} \sum_{k=1}^{K} r_{ik} \log(p^B(r_{ik} = 1|s_i, o_i)) + (1 - r_{ik}) \log(1 - p^B(r_{ik} = 1|s_i, o_i)) \)
       16. \( L_{\text{REMAP-B}} = L^T + L^G + \lambda_B [L^B + D_{KL}(p^B, p^T) + D_{KL}(p^B, p^G)] \)
       17. \( \Theta \leftarrow \text{Update}(\Theta, L_{\text{REMAP-B}}) \)

graph together with text descriptions retrieved from medical data repositories using distant supervision. Further, we processed a large electronic health record dataset and curated a high quality human annotated dataset.

Disease knowledge graph. We construct a disease-disease knowledge graph from Diseases [58] and MedScape [59] repositories that unify evidence on disease-disease associations from automatic text mining, manually curated literature, cancer mutation data, and genome-wide association studies. This knowledge graph contains 9,182 disease concepts, which are assigned concept unique identities (CUI) in Unified Medical Language System (UMLS), and three types of relationships between disease concepts, which are ‘may cause’ (MC), ‘may be caused by’ (MBC), and ‘differential
diagnosis’ (DDx). Other relations in the knowledge graph are denoted as ‘not available’ relations (NA). The MBC relation type is the reverse relation of the MC relation type while DDx is a symmetric relation between disease concepts. Further dataset statistics are in [9] and Table 1.

**Language dataset.** We use a text corpus taken from Lin et al. [9] that includes sentences from Wikipedia, Medscape eMedicine, PubMed abstracts from The PubMed Baseline Repository provided by the National Library of Medicine, and four textbooks. This dataset comprises 21,315,153 medical articles, and altogether 1,466,065 sentences from these articles can be aligned to disease-disease edges in the disease knowledge graph. Details are shown in Table 1.

**Electronic health record dataset.** We use two types of information from electronic health records (EHRs), both taken from Beam et al. [60]. In particular, using 20 million medical notes, Beam et al. [60] created a dataset of disease concepts that appear together in the same note and used the SVD to decompose the resulting co-occurrence matrix and produce embedding vectors for disease concepts. We use information on co-occurring disease concepts to guide sampling of negative node pairs when training the disease knowledge graph. Further, we use embedding vectors from [60] to initialize embeddings in REMAP’s graph neural network. For out-of-dictionary disease concepts, we use the average of all concept embedding as their initial embeddings.

**Human annotated dataset.** We randomly selected 500 disease pairs to create an annotated dataset. For that, we recruited three senior physicians from Peking Union Medical College Hospital who manually labeled disease pairs. Annotation experts disagreed on labels for 14 disease pairs (i.e., 2.8% of the total number of disease pairs) that were resolved through consensus voting. The human annotated dataset is used for model comparison and performance evaluation.

3.2 Training REMAP models

Next we outline training details of REMAP models, including negative sampling, pre-training strategy for language and graph models, and the multimodal learning approach.

**Negative sampling.** We construct negative samples by sampling disease pairs whose co-occurrence in the EHR co-occurrence matrix [60] is lower than a pre-defined threshold. In particular, we expect that two unrelated diseases rarely appear together in EHRs, meaning that the corresponding values in the co-occurrence matrix are low. Thus, such disease pairs represent
Table 1: Overview of the disease knowledge graph and the language dataset. Shown are statistics for the following relation types: Not Available (NA), Differential Diagnosis (DDx), May Cause (MC), and May Be Caused by (MBC). Total denotes the total number of all triplets (see Figure 2).

| Dataset   | Total | NA    | DDx   | MC   | MBC   | Entities |
|-----------|-------|-------|-------|------|-------|----------|
| Unaligned |       |       |       |      |       |          |
|           | 96,913| 30,546| 20,657| 23,411| 22,298| 9,182    |
| Aligned   |       |       |       |      |       |          |
| Train     |       |       |       |      |       |          |
| Triplet   | 31,037| 15,670| 7,262 | 4,358| 3,747 | 7,794    |
| Language  | 1,244,874| 799,194| 208,921| 123,735| 113,024|          |
| Validation|       |       |       |      |       |          |
| Triplet   | 7,754 | 3,918 | 1,821 | 1,065| 950   | 4,433    |
| Language  | 206,179| 68,934| 60,165| 43,706| 33,474|          |
| Annotated |       |       |       |      |       |          |
| Triplet   | 499   | 8     | 210   | 159  | 122   | 733      |
| Language  | 15,012| 96    | 4,980 | 6,699| 3,237 |          |

suitable negative samples to train models for classifying disease-disease relations.

**Pre-training a language model.** The text-based model comprises the text encoder and the TuckER module for disease-disease relation prediction. We denote the relation embeddings as \( r^T \), and the loss function as \( L^T \) (Section 2.3). In particular, we use SciBERT tokenizer and SciBERT-SciVocab-uncased model [61]. The entity markers are added to the SciVocab vocabulary, and their embeddings are initialized with uniform distribution. We set the maximum number of sentences in a bag to \( l_{m(\text{max})} \). If the bag size is greater than \( l_{m(\text{max})} \), then \( l_{m(\text{max})} \) sentences are selected uniformly at random for training. Further details on hyper-parameters are in Table A4.

**Pre-training a graph neural network model.** The graph-based model comprises the heterogeneous attention network encoder and the TuckER module for disease-disease relation prediction. The relation embeddings produced by the TuckER module are denoted as \( r^G \). In the pre-training phase, the model is optimized for the loss function is \( L^G \) (Section 2.3). The initial embeddings for nodes in the disease knowledge graph are concept unique identifier (CUI) representations derived from the SVD decomposition of the EHR co-occurrence matrix [60]. Further details on hyper-parameters are in Table A4.

**Joint learning on the multimodal dataset.** After data type-specific pre-training is completed, the text and graph models are fused in cross-modal learning. To this end, the shared relation vector \( r \) is initialized as: \( r = (r^T + r^G)/2 \). We consider two REMAP variants, namely REMAP-M and REMAP-B, that are optimized for different loss functions (Section 2.3). Details on hyper-parameter selection are in Table A4.

13
3.3 Experimental setup

Next we overview performance metrics, baseline methods, and variants of REMAP approach.

**Baseline methods and performance metrics.** We consider 11 baseline methods for disease relation extraction, including 5 knowledge graph-based methods and 6 text-based methods. Detailed descriptions of all methods are in Appendix A. We evaluate predicted disease-disease relations by calculating the accuracy of predicted relations between disease concepts, which is an established approach for benchmarking relation extraction methods [62]. Specifically, given a triplet \((s_i, r_i, o_i)\) and a predicted score \(p_i \in [0, 1]\), relation \(r_i\) is predicted to exist between \(s_i\) and \(o_i\) if the predicted score \(p_i \geq \text{threshold}_i\), which corresponds to a binary classification task for each relation type. The threshold \(i\) is a relation type-specific value determined such that binary classification performance achieves maximal F1-score. We report classification accuracy, precision, recall, and F1-score for all experiments in this study.

**Variants of REMAP approach.** We carry out an ablation study to examine the utility of key REMAP components. We consider the following three components and examine REMAP’s performance with and without each.

- **REMAP-B without joint learning** In text-only ablations, we use SciBERT to obtain concept embeddings \(h^T_s\) and \(h^T_o\), and combine them to produce a relation embedding \(r^k\), where \(r^k\) is related to the concept embeddings based on text information. Finally, we consider three scoring functions to classify disease-disease relations, and we denote the models as SciBERT (linear), SciBERT (TransE), and SciBERT (TuckER). Similarly, in graph-only ablations, we first use a heterogeneous attention network to obtain graph embeddings \(h^G_s\) and \(h^G_o\) that are combined into prediction by different scoring functions, including HAN (linear), HAN (TransE), and HAN (TuckER).

- **REMAP-B without EHR embeddings** REMAP-B uses EHR embeddings [60] as initial node embedding \(H^{init}\). To examine the utility of EHR embeddings, we design an ablation study that initializes node embeddings using the popular Xavier initialization [63] instead of EHR embeddings. Other parts of the model are the same as in REMAP-B.

- **REMAP-B without unaligned triplets** Unaligned triplets denote triplets in the disease knowledge graph that do not have the corresponding sentences in the language dataset. To
demonstrate how these unaligned triplets influence model performance, we design an ablation study in which we train a REMAP-B model on the reduced disease knowledge graph with unaligned triplets excluded.

4 RESULTS

REMAP is a multimodal approach for joint learning on graph and text datasets. We evaluate REMAP’s prowess for disease relation extraction when the REMAP model is tasked to identify candidate disease relations in either text (Section 4.1) or graph-structured (Section 4.2) data. We present ablation (Appendix B) and case (Section 5) studies.

4.1 Extracting disease relations from text

We start with results on the human annotated dataset where each method, while it can be trained on a multimodal text-graph dataset, is tasked to extract disease relations from text alone, meaning that the test set consists of only annotated sentences. Table 2 shows performance on the annotated set for text-based methods. Neural network models outperform methods, such as Naive Bayes and SVM. Further, BiGRU+Attention is the best performing baseline method, achieving an accuracy of 78.6 and an F1-score of 64.6. We also find that REMAP-B and REMAP-M achieve the best performance across all settings, outperforming baselines significantly. In particular, REMAP models surpass the strongest baseline by 10.0 absolute percentage points (accuracy) and by 7.2 absolute percentage points (F1-score). These results show that multimodal learning can considerably advance disease relation extraction when only one data type is available at test time.

4.2 Identifying disease relations in a graph-structured dataset

We proceed with results of disease relation prediction in a setting where each method is tasked to classify disease relations based on the disease knowledge graph alone. This setting evaluates the flexibility of REMAP as REMAP can answer either graph-based or text-based disease queries. In particular, Table 2 shows performance results attained on the human annotated dataset with query disease pairs given only as disease concept nodes in the knowledge graph. We find that graph-based baselines perform better than text-based baselines. Further, TuckER, the best
Table 2: Results of disease relation extraction on the human annotated set. DDx: differential diagnosis, MC: may cause, MBC: may be caused by. The “micro” columns denote micro average accuracy or F1-score for DDx, MC, and MBC relation types. Further results are in Appendix C.

| Modality | Model           | Accuracy | F1-score |
|----------|-----------------|----------|----------|
|          | micro DDx MC MBC |          |          |
|          | micro DDx MC MBC |          |          |
| Text     | SVM             | 59.5     | 59.0     | 68.2     | 51.2     | 31.5     | 40.9     | 25.4     | 19.0     |
| Baselines| NB              | 74.3     | 71.4     | 69.0     | 82.4     | 47.1     | 60.2     | 29.9     | 43.2     |
|          | RF              | 72.0     | 68.6     | 70.8     | 76.4     | 37.8     | 53.1     | 25.5     | 19.2     |
|          | TextCNN         | 76.7     | 75.4     | 73.0     | 81.8     | 60.9     | 67.5     | 59.9     | 48.6     |
|          | BiGRU           | 77.4     | 73.0     | 77.2     | 82.0     | 62.0     | 67.9     | 54.0     | 59.8     |
|          | BiGRU+attention | 78.6     | 75.0     | 78.2     | 82.6     | 64.6     | 67.7     | 63.5     | 60.6     |
|          | REMAP           | 88.2     | 83.6     | 89.0     | 92.0     | 80.9     | 80.7     | 80.0     | 82.6     |
| Ours     | REMAP-M         | 88.6     | 84.2     | 89.0     | 92.8     | 81.5     | 81.6     | 79.6     | 83.8     |
|          | REMAP-B         | 88.6     | 84.4     | 89.2     | 92.4     | 81.8     | 81.9     | 80.3     | 83.3     |
| Graph    | TransE_l2       | 75.1     | 70.7     | 72.7     | 81.8     | 63.2     | 68.0     | 57.0     | 62.2     |
| Baselines| DistMult        | 69.8     | 77.5     | 61.3     | 70.5     | 56.1     | 71.0     | 43.4     | 51.5     |
|          | ComplEx         | 79.0     | 75.2     | 77.8     | 84.2     | 65.0     | 69.3     | 56.5     | 66.9     |
|          | RGCN            | 71.8     | 78.6     | 62.5     | 74.3     | 62.2     | 75.1     | 50.8     | 58.6     |
|          | TuckER          | 81.5     | 77.6     | 82.3     | 84.7     | 73.7     | 76.2     | 71.7     | 71.9     |
|          | REMAP           | 89.6     | 86.4     | 89.6     | 92.8     | 83.5     | 84.3     | 81.6     | 84.2     |
| Ours     | REMAP-M         | 89.3     | 87.0     | 88.4     | 92.6     | 83.3     | 85.7     | 78.8     | 83.8     |
|          | REMAP-B         | **89.8** | **87.3** | **89.9** | 92.2     | **84.1** | **85.8** | **82.4** | 82.7     |

The graph-based method, outperforms BiGRU+Attention, a top-performing text-based baseline, by 2.9 (accuracy) and 8.9 absolute percentage points (F1-score). These results suggest that the disease knowledge graph contains more valuable predictive information than text when classifying disease-disease relations. Last, we find that REMAP-B is a top performer among REMAP variants, achieving an accuracy of 89.8 and an F1-score of 84.1. Further results are in Appendices B-D.

5 DISCUSSION

Table A1 shows that joint learning can considerably improve model performance in both text and graph modalities. To further analyze joint learning in REMAP-B, we analyze the characteristics of correctly and incorrectly classified disease-disease relationships. Comparing full REMAP-B with REMAP-B w/o joint learning (Table A1), we found that 28 predicted disease-disease relationships turned from incorrect to correct predictions. Conversely, we found that 14 predicted disease-disease relationships turned correct to incorrect predictions on the text modality. Besides,
19 predictions turned from incorrect to correct and 11 inversely on the graph modality. It can be seen that improved classification accuracy on the text modality is why REMAP-B achieved better performance on the annotated set.

We proceed with a case study examining the relationship between hypogonadism and Goldberg-Maxwell syndrome. Hypogonadism and Goldberg-Maxwell syndrome are separate and distinct diseases [64, 65]. However, many physicians fail correctly to differentiate in their diagnosis of these two diseases, especially in light of Goldberg-Maxwell syndrome being a rare disease. Differential diagnosis is a process wherein a doctor differentiates between two or more conditions that could be behind a person’s symptoms.

Figure 3 illustrates how the prediction of the “differential diagnosis” (DDx) relationship between hypogonadism and Goldberg-Maxwell syndrome changed from incorrect to correct after joint learning with the text modality. Because REMAP-B correctly recognizes diseases from text and unifies them with diseases in the knowledge graph, it can be seen that both diseases have rich local network neighborhoods in the graph. For example, hypogonadism and Goldberg-Maxwell syndrome have many outgoing edges of type differential diagnosis. Further, meta-paths connect these diseases in the graph, such as as second-order “differential diagnosis $\rightarrow$ differential diagnosis” meta-path, and their outgoing degrees in the graph are relatively high. In the case of joint learning, the text-based model can extract part of the disease representation from the graph modality to update its internal representations and thus improve text-based classification of relations.

6  CONCLUSION

We develop REMAP, a multi-modal learning approach for disease relation extraction that integrates knowledge graphs and language models. Results on a dataset annotated by experts show that REMAP considerably outperforms methods for learning on text or knowledge graphs alone. Further, REMAP can extract and classify disease relationships in the most challenging settings where text or graph information is absent. Finally, we provide a new data resource of extracted relationships between diseases that can serve as a benchmarking dataset for systematic evaluation and comparison of disease relation extraction algorithms.
Funding and acknowledgments

We gratefully acknowledge the support of the Harvard Translational Data Science Center for a Learning Healthcare System (CELeHS). M.Z. is supported, in part, by NSF under nos. IIS-2030459 and IIS-2033384, US Air Force Contract No. FA8702-15-D-0001, Harvard Data Science Initiative, Amazon Research Award, Bayer Early Excellence in Science Award, AstraZeneca Research, and Roche Alliance with Distinguished Scientists Award. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

Author contributions

Y.L and K.L. are co-first authors and have developed the method and produced all analyses presented in the paper. S.Y. contributed experimental data. T.C. and M.Z. conceived and designed the study. The project was guided by M.Z., including designing methodology and outlining experiments. Y.L., K.L., S.Y., T.C., and M.Z. wrote the manuscript. All authors discussed the results and reviewed the paper.

Data and code availability

Python implementation of REMAP is available on Github at https://github.com/Lukeming-tsinghua/REMOD. The human annotated dataset used for evaluation of REMAP is available at https://doi.org/10.6084/m9.figshare.17776865.

Conflicts of interest

None declared.
Figure 1: Overview of REMAP architecture. REMAP introduces a novel co-training learning strategy that continually updates a multimodal language-graph model for disease relation extraction and classification. Language and graph encoders specify deep transformation functions that embed disease concepts (i.e., subject entities s and object entities o) from the language data $D^T$ and disease knowledge graph $D^G$ into compact embeddings, producing condensed summaries of language semantics and biomedical knowledge for every disease. Embeddings output by the encoders (i.e., $h^s_T$, $h^o_T$, $h^s_G$, $h^o_G$) are then combined in a disease relation type-specific manner (e.g., “differential diagnosis” and “may cause” relation types) and passed to a scoring function that calculates the probability representing how likely two diseases are related to each other and what kind of relationship exists between them.

Figure 2: Creating a dataset split for performance evaluation and benchmarking. Triplets in the disease knowledge graph are divided into aligned triplets and unaligned triplets. Aligned triplets are triplets that have corresponding sentences in the language dataset. Unaligned triplets have no corresponding sentences in the language dataset.
7 in 48 subjects (24 couples) with sterility (14.5%), 11 in 96 with hypogonadism and/or cryptorchidism (11.4%), 15 in 38 women with oligoamenorrhea (39.4%) and 4 in 4 patients with Goldberg-Maxwell syndrome (100%).

Endocrinologically, these patients often have the features of hypogonadism associated to the Goldberg-Maxwell syndrome, particularly its partial forms.

Figure 3: Illustration of (hypogonadism, DDx, Goldberg-Maxwell syndrome) triplet in REMAP. This triplet represents the relationship between symptomatically similar diseases, hypogonadism and Goldberg-Maxwell syndrome. The subject entity (hypogonadism, C0020619) is shown in orange and the object entity (Goldberg-Maxwell syndrome, C0936016) is shown in blue. (a) The subject and object entities are identified with unique UMLS concept identifiers. The relation between them is differential diagnosis. There is a DDx-DDx meta-path between these entities and that information is leveraged by our graph encoder to predict the relationship between hypogonadism and Goldberg-Maxwell syndrome. The bar plots indicate distribution of relation types going out of the subject or object entities in the knowledge graph. (b) We show two sentences representing the triplet. We use <sep> token to separate the sentences and the title of article from which we mine the sentence. If the article is from PubMed, we use special token <empty_title> as a placeholder. We add two special tokens before and after terms in each sentence to identify the positions of entities. For example, we add <entity-t047-1> before hypogonadism to mark the beginning of subject entity; t047 serves as a identifier of the semantic type for hypogonadism.
References

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In ACM SIGMOD, pages 1247–1250, 2008.

[2] Thomas C Rindflesch, Halil Kilicoglu, Marcelo Fiszman, Graciela Rosemblat, and Dongwook Shin. Semantic medline: An advanced information management application for biomedicine. Information Services & Use, pages 15–21, 2011.

[3] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In AAAI, 2010.

[4] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In ACM SIGKDD, pages 601–610, 2014.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, pages 4171 – 4186, 2019.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In EMNLP-IJCNLP, pages 3615–3620, 2019.

[7] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In ACL, pages 2895–2905, 2019.

[8] Zexuan Zhong and Danqim Chen. A frustratingly easy approach for joint entity and relation extraction. In NAACL-HT, pages 50–61, 2021.

[9] Yucong Lin, Keming Lu, Yulin Chen, Chuan Hong, and Sheng Yu. High-throughput relation extraction algorithm development associating knowledge articles and electronic health records. arXiv:2009.03506, 2020.

[10] Jing Chen, Baotian Hu, Weihua Peng, Qingcai Chen, and Buzhou Tang. Biomedical relation extraction via knowledge-enhanced reading comprehension. BMC Bioinformatics, pages 1–19, 2022.

[11] Gang Li, Cathy Wu, and K Vijay-Shanker. Noise reduction methods for distantly supervised biomedical relation extraction. In BioNLP, pages 184–193, 2017.

[12] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12):2724–2743, 2017.

[13] Michelle M Li, Kexin Huang, and Marinka Zitnik. Representation learning for networks in biology and medicine: advancements, challenges, and opportunities. arXiv:2104.04883, 2021.

[14] Marinka Zitnik and Blaz Zupan. Collective pairwise classification for multi-way analysis of disease and drug data. In the Pacific Symposium on Biocomputing, pages 81–92, 2016.

[15] Baoxu Shi and Tim Weninger. ProjE: Embedding projection for knowledge graph completion. In AAAI, volume 31, 2017.

[16] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In AAAI, 2015.

[17] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: Knowledge graph attention network for recommendation. In KDD, pages 950–958, 2019.

[18] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. Recurrent knowledge graph embedding for effective recommendation. In ACM Conference on Recommender Systems, pages 297–305, 2018.
[19] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, 26:2787–2795, 2013.

[20] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *EMNLP*, pages 1591–1601, 2014.

[21] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, pages 687–696, 2015.

[22] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, 2011.

[23] B. Yang, Wen tau Yih, X. He, Jianfeng Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. *Computing Research Repository*, abs/1412.6575, 2015.

[24] Théo Trouillon, Christopher R Dance, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *JMLR*, 18, 2017.

[25] Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *EMNLP-IJCNLP*, pages 5185–5194, 2019.

[26] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[27] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. Relational graph attention networks. *arXiv:1904.05811*, 2019.

[28] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[29] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

[30] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP*, pages 1003–1011, 2009.

[31] Ivana Balazevic, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *EMNLP*, 2019.

[32] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-BERT: Enabling language representation with knowledge graph. In *AAAI*, volume 34, pages 2901–2908, 2020.

[33] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: Enhanced representation through knowledge integration. *arXiv:1904.09223*, 2019.

[34] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *ACL*, pages 1441–1451, 2019.

[35] Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of EMNLP*, pages 2281–2290, 2020.

[36] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. *arXiv:1904.02342*, 2019.
[37] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. Colake: Contextualized language and knowledge embedding. In International Conference on Computational Linguistics, pages 3660–3670, 2020.

[38] Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. Improving distantly-supervised relation extraction with joint label embedding. In EMNLP, pages 3812–3820, 2019.

[39] Peng Xu and Denilson Barbosa. Connecting language and knowledge with heterogeneous representations for neural relation extraction. NAACL-HLT, page 3201–3206, 2019.

[40] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In NAACL-HLT, pages 3016–3025, 2019.

[41] Yashen Wang, Huanhuan Zhang, Ge Shi, Zhirun Liu, and Qiang Zhou. A model of text-enhanced knowledge graph representation learning with mutual attention. IEEE Access, 8:52895–52905, 2020.

[42] Xu Han, Zhiyuan Liu, and Maosong Sun. Joint representation learning of text and knowledge for knowledge graph completion. arXiv:1611.04125, 2016.

[43] Zizheng Ji, Zhengchao Lei, Tingting Shen, and Jing Zhang. Joint representations of knowledge graphs and textual information via reference sentences. IEICE Transactions on Information and Systems, pages 1362–1370, 2020.

[44] Qin Dai, Naoya Inoue, Paul Reisert, Ryo Takahashi, and Kentaro Inui. Distantly supervised biomedical knowledge acquisition via knowledge graph based attention. In the Workshop on Extracting Structured Knowledge from Scientific Publications, pages 1–10, 2019.

[45] George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. Improving relation extraction by leveraging knowledge graph link prediction. arXiv:2012.04812, 2020.

[46] Zhengqiu He, Wenliang Chen, Yuyi Wang, Wei Zhang, Guanchun Wang, and Min Zhang. Improving neural relation extraction with positive and unlabeled learning. In AAAI, volume 34, pages 7927–7934, 2020.

[47] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. IEEE Access, 2020.

[48] Peng Su, Yifan Peng, and K. Vijay-Shanker. Improving bert model using contrastive learning for biomedical relation extraction. In BIONLP, 2021.

[49] Qi Wang, Liang Zhan, Paul Thompson, and Jiayu Zhou. Multimodal learning with incomplete modalities by knowledge distillation. In SIGKDD, page 1828–1838, 2020.

[50] Qiuling Suo, Weida Zhong, Fenglong Ma, Ye Yuan, Jing Gao, and Aidong Zhang. Metric learning on healthcare data with incomplete modalities. In IJCAI, pages 3534–3540, 2019.

[51] Tao Zhou, Mingxia Liu, Kim-Han Thung, and Dinggang Shen. Latent representation learning for alzheimer’s disease diagnosis with incomplete multi-modality neuroimaging and genetic data. IEEE Transactions on Medical Imaging, pages 2411–2422, 2019.

[52] Yang Yang, De-Chuan Zhan, Xiang-Rong Sheng, and Yuan Jiang. Semi-supervised multi-modal learning with incomplete modalities. In IJCAI, pages 2998–3004, 2018.

[53] Lei Cai, Zhengyang Wang, Hongyang Gao, Dinggang Shen, and Shuiwang Ji. Deep adversarial learning for multi-modality missing data completion. In ACM SIGKDD, pages 1158–1166, 2018.

[54] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In ACII, pages 202–208, 2017.
[55] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. PathSim: meta path-based top-k similarity search in heterogeneous information networks. In VLDB, pages 992–1003, 2011.

[56] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In NeurIPS, 2018.

[57] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In CVPR, pages 11020–11029, 2020.

[58] Sune Pletscher-Frankild, Albert Pallejà, Kalliopi Tsafou, Janos X Binder, and Lars Juhl Jensen. DISEASES: text mining and data integration of disease–gene associations. Methods, 74:83–89, 2015.

[59] Peter Frishauf. Medscape—the first 5 years. Medscape General Medicine, page 5, 2005.

[60] Andrew L Beam, Benjamin Kompa, Allen Schmultz, Inbar Fried, Griffin Weber, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. In Pacific Symposium on Biocomputing 2020, pages 295–306, 2019.

[61] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. arXiv:1910.03771, 2019.

[62] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for machine learning on graphs. NeurIPS, 2020.

[63] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, pages 249–256, 2010.

[64] Ba D Nguyen. PET imaging of sertoli cell tumor in androgen insensitivity syndrome. Clinical Nuclear Medicine, pages 743–745, 2003.

[65] Monika Grymowicz, Ewa Rudnicka, Katarzyna Smolarczyk, Roman Smolarczyk, Anna Szeliga, Agnieszka Podfigurna, and Błażej Męczekalski. Complete androgen insensitivity syndrome. Practical Clinical Endocrinology, pages 425–431, 2021.

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

[67] Theo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. ICML, 2016.

[68] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. arXiv:1703.06103, 2017.

[69] Yoon Kim. Convolutional neural networks for sentence classification. In EMNLP, pages 1746–1751, 2014.
Appendix

A Further information on baseline methods

We consider 11 baseline methods divided into two groups: 5 methods for link prediction in knowledge graphs and 6 methods for relation extraction in text.

Graph-based methods are trained on the disease knowledge graph and include the following:

- **TransE** [19] predicts disease-disease relationships by translating embedding vectors in the learned embedding space. Given embeddings of a triplet \((h, r, t)\), the score is calculated as
  \[ s = ||h + r - t||^2_2. \]
  We train our TransE model using negative sampling and \(L2\) penalty as recommended by the authors.

- **DistMult** [66] predicts disease-disease relationships using a bilinear decoder for edge in the knowledge graph. Given embeddings of a triplet \((h, r, t)\), the score is calculated as \(s = h M_r t\), where \(M_r\) is a trainable diagonal matrix for relation \(r\).

- **ComplEx** [67] predicts disease-disease relationships by carrying out a latent factorization using complex valued embeddings. Given embeddings of a triplet \((h, r, t)\), the score is calculated as
  \[ s = Re(<h, r, t>). \]
  The product \(<\cdot>\) is Hermitian product and relation embedding \(r\) is a complex-valued vector.

- **RGCN** [68] predicts disease-disease relationships using relational graph convolutional network (RGCN). Following authors’ recommendations, We use a 2-layer RGCN to embed the knowledge graph and DistMult decoder for link prediction. We train the RGCN model using the cross-entropy loss and Adam optimizer.

- **TuckER** [31] is a knowledge graph embedding method that uses Tucker decomposition. Given embeddings of a triplet \((h, r, t)\), the score is calculated as \(s = W \times_1 h \times_2 r \times_3 t\), where \(W\) is a trainable matrix and \(\times_i\) denotes tensor multiplication on dimension \(i\).
Text-based methods are trained on the language dataset following the strategy in Lin et al. [9]. We consider two types of text features: a) bag-of-words (BoW) representations given by the N-grams filtered using the TF-IDF technique, and b) embedding vectors generated by a skip-gram model. In sentence-level baselines, including support vector machines, naive Bayes, random forest, TextCNN, and BiGRU, we use the voting strategy on sentence-level predictions to determine the final prediction for input examples that are longer than one sentence. We consider the following text-based methods:

- **Support Vector Machines (SVM)** use scaled BoW features with linear kernel as the classifier.
- **Naive Bayes (NB)** uses scaled BoW features as input to a multinomial Naive Bayes classifier with a smooth factor $\alpha$ set to $\alpha = 0.1$.
- **Random Forest (RF)** uses scaled BoW features with 100 decision trees and the Gini criterion as the classifier.
- **TextCNN** [69] is a convolutional neural network for text classification. We use skip-gram embeddings to initialize the TextCNN model and train it using binary cross-entropy loss and Adam optimizer.
- **BiGRU** is a recurrent neural network initialized in the same way as TextCNN. The encoder is a 1-layer BiGRU whose outputs are aggregated into embeddings using a one-headed self-attention. The final embeddings are fed into a linear layer optimized for sentence classification.
- **BiGRU+Attention** is the same as the BiGRU baseline with an added instance-level attention. Instead of voting, this approach uses instance-level attention to aggregate latent vectors across all sentences in an instance.

We use grid search on the validation set to select hyper-parameters for all methods.

Table A4 outlines the hyper-parameter selection in REMAP.
Table A1: Results of ablation study. DDx: differential diagnosis, MC: may cause, MBC: may be caused by. The “micro” columns denote micro average accuracy or F1-score for DDx, MC, and MBC relation types.

| Modality | Model                              | Accuracy   | F1-score   |
|----------|------------------------------------|------------|------------|
|          |                                    | micro DDx | MC MBC     | micro DDx | MC MBC     |
| Text     | REMAP-B                            | 88.6       | 84.4 89.2  | 92.4       | 81.8       | 81.9 80.3  | 83.3       |
|          | w/o joint learning (linear)        | 87.3       | 84.1 86.3  | 91.3       | 78.9       | 81.2 73.4  | 80.9       |
|          | w/o joint learning (TransE)        | 86.1       | 79.1 89.1  | 90.2       | 80.5       | 79.3 82.2  | 80.8       |
|          | w/o joint learning (TuckER)        | 87.9       | 83.6 87.0  | **93.2**   | 80.0       | 80.1 75.7  | 85.1       |
|          | w/o EHR embedding                  | 87.6       | 83.2 88.2  | 91.4       | 79.6       | 79.8 78.2  | 81.1       |
|          | w/o unaligned triplets             | 88.2       | 83.0 88.5  | **93.2**   | 81.0       | 80.0 78.8  | **85.2**   |
| Graph    | REMAP-B                            | **89.8**   | **87.3**  | **89.9**   | **92.2**   | **84.1**  | **85.8**  | **82.4**  |
|          | w/o joint learning (linear)        | 87.4       | 82.4 89.3  | 90.2       | 80.5       | 80.5 82.1  | 78.4       |
|          | w/o joint learning (TransE)        | 85.8       | 81.6 85.3  | 90.4       | 77.2       | 78.9 73.1  | 78.9       |
|          | w/o joint learning (TuckER)        | 88.9       | 85.8 89.4  | 91.6       | 82.5       | 83.9 81.0  | 81.4       |
|          | w/o EHR embedding                  | 87.6       | 84.4 87.2  | 91.4       | 80.3       | 82.0 76.8  | 81.4       |
|          | w/o unaligned triplets             | 87.3       | 84.8 87.8  | 89.4       | 79.5       | 82.4 77.7  | 76.2       |

B Results: Ablation study

Table A1 shows results of ablation study. In both modalities, we see that model performance without EHR embeddings is 1-3 absolute percentage points worse than that of the proposed approach. This result provides evidence that EHR embeddings can have a significant role in classifying disease-disease relationships. Similarly, REMAP-B without unaligned triplets in the knowledge graph performs worse than REMAP-B, indicating the importance of leveraging unsupervised information from unaligned triplets. We further probe REMAP in Appendix D and examine the utility of EHR embeddings in training the REMAP models.

C Further information on REMAP’s performance

Table A2 extends Table 2 from the main text with additional information about REMAP’s performance measured using precision, recall, and F1-score metrics. Shown is performance that REMAP achieves when the REMAP model, although trained on the multimodal graph-text dataset, is asked to make predictions at test time using only text information.
Table A2: Performance of multimodal methods for identifying candidate disease-disease relations from text data. Shown is average performance across multiple independent runs calculated on the human annotated set. Higher values indicate better performance.

| Model               | Precision minor | Precision DDx | Precision MC | Precision MBC | Recall minor | Recall DDx | Recall MC | Recall MBC | F1-score minor | F1-score DDx | F1-score MC | F1-score MBC |
|---------------------|-----------------|---------------|--------------|--------------|--------------|------------|-----------|------------|----------------|--------------|-------------|--------------|
| SVM                 | 50.0            | 52.2          | 50.0         | 41.7         | 23.0         | 33.6       | 17.0      | 12.3       | 31.5           | 40.9         | 25.4        | 19.0         |
| Naive Bayes         | 69.1            | 73.0          | 53.2         | 75.0         | 35.7         | 51.2       | 20.8      | 30.4       | 47.1           | 60.2         | 29.9        | 43.2         |
| Random Forest       | 69.2            | 71.8          | 67.6         | 58.3         | 26.0         | 42.2       | 15.7      | 58.3       | 37.8           | 53.1         | 25.5        | 19.2         |
| TextCNN             | 67.8            | 76.2          | 56.7         | 78.2         | 55.3         | 60.7       | 63.5      | 35.2       | 60.9           | 67.5         | 59.9        | 48.6         |
| BiGRU               | 69.1            | 68.1          | 75.3         | 65.7         | 56.3         | 67.8       | 42.1      | 54.9       | 62.0           | 67.9         | 54.0        | 59.8         |
| BiGRU + attention   | 70.6            | 74.4          | 67.9         | 67.7         | 59.6         | 62.1       | 59.7      | 54.9       | 64.6           | 67.7         | 63.5        | 60.6         |
| SciBERT (linear)    | 86.6            | 81.0          | 96.9         | 88.3         | 72.5         | 81.4       | 59.1      | 74.6       | 78.9           | 81.2         | 73.4        | 80.9         |
| SciBERT (TransE)    | 74.9            | 68.2          | 86.2         | 77.4         | 87.0         | 94.8       | 78.6      | 84.4       | 80.5           | 79.3         | 82.2        | 80.8         |
| SciBERT (TuckER)    | 87.3            | 81.7          | 93.5         | 91.5         | 73.9         | 78.6       | 63.5      | 79.5       | 80.0           | 80.1         | 75.7        | 85.1         |
| REMAP               | 85.8            | 79.9          | 94.8         | 88.0         | 76.6         | 81.4       | 69.2      | 77.9       | 80.9           | 80.7         | 80.0        | 82.6         |
| REMAP-M             | 87.4            | 79.9          | 97.3         | 93.0         | 76.4         | 83.3       | 67.3      | 76.2       | 81.5           | 81.6         | 79.6        | 83.8         |
| REMAP-B             | 86.2            | 79.7          | 95.7         | 89.6         | 77.8         | 84.3       | 69.2      | 77.9       | 81.8           | 81.9         | 80.3        | 83.3         |

Table A3 extends Table 2 from the main text with additional information about REMAP’s performance measured using precision, recall, and F1-score metrics. Shown is performance that REMAP achieves when the REMAP model, although trained on the multimodal graph-text dataset, is asked to make predictions at test time using only information from the disease knowledge graph.

D Results: Further analysis of REMAP

We hypothesize that the construction of negative samples based on the prevalence of disease-disease co-mentions in an EHR system can yield more representative negative samples than alternative approaches, which typically randomly sample negative samples with no consideration of prior knowledge, e.g., [9]. Negative samples in REMAP are constructed based on the co-occurrence matrix of disease concepts in electronic health records. In particular, a disease-disease pair constitutes a good negative sample if the two diseases rarely occur together in EHRs. As a result, negative samples in REMAP better mimic real-world medical evidence. Further, we can impose a threshold on the co-occurrence matrix to reduce the false-negative rate. Finally, negative samples
Table A3: Performance of multimodal methods for identifying candidate disease-disease relations from graph-structured data. Shown is average performance across multiple independent runs calculated on the human annotated set. Higher values indicate better performance.

| Model        | Precision | Recall | F1-score |
|--------------|-----------|--------|----------|
|              | minor DDx | minor DDx | minor DDx | minor DDx |
| TransE_l2    | 61.3      | 63.0   | 57.3      | 63.0      | 65.2      | 73.8   | 56.6      | 61.5      | 63.2      | 68.0   | 57.0      | 62.2      |
| DistMult     | 53.6      | 77.8   | 40.7      | 43.1      | 58.9      | 65.2   | 46.5      | 63.9      | 56.1      | 71.0   | 43.4      | 51.5      |
| ComplEx      | 71.7      | 72.2   | 75.0      | 68.4      | 59.5      | 66.7   | 54.3      | 65.6      | 65.0      | 69.3   | 56.5      | 66.9      |
| RGCN         | 56.2      | 74.9   | 44.2      | 48.9      | 69.7      | 75.2   | 59.7      | 73.0      | 62.2      | 75.1   | 50.8      | 58.6      |
| TuckER       | 70.1      | 69.8   | 74.3      | 66.2      | 77.8      | 83.8   | 69.2      | 78.7      | 73.7      | 76.2   | 71.7      | 71.9      |
| HAN (linear) | 82.0      | 75.2   | 92.9      | 84.8      | 79.0      | 86.7   | 73.6      | 73.0      | 80.5      | 80.5   | 82.1      | 78.4      |
| HAN (TransE) | 81.3      | 76.1   | 88.4      | 84.9      | 73.5      | 81.9   | 62.3      | 73.8      | 77.2      | 78.9   | 73.1      | 78.9      |
| HAN (TuckER) | 85.7      | 80.1   | 94.2      | 88.5      | 79.4      | 88.1   | 71.1      | 75.4      | 82.5      | 83.9   | 81.0      | 81.4      |
| REMAP        | 87.0      | 81.7   | 93.5      | 90.6      | 80.2      | 87.1   | 72.3      | 78.7      | 83.5      | 84.3   | 81.6      | 84.2      |
| REMAP-M      | 85.6      | 79.8   | 93.9      | 89.7      | 81.1      | 92.4   | 67.9      | 78.7      | 83.3      | 85.7   | 78.8      | 83.8      |
| REMAP-B      | 86.6      | 81.3   | 93.6      | 90.3      | 81.7      | 91.0   | 73.6      | 76.2      | 84.1      | 85.8   | 82.4      | 82.7      |
Table A4: Selection of hyper-parameters in REMAP. We use grid search to select hyper-parameters on the validation set.

| REMAP’s component | Model parameter                          | Notation | Value  |
|-------------------|------------------------------------------|----------|--------|
| Neural architecture | Padding length of sentences               | $d_t$     | 256    |
|                    | Hidden size of SciBERT output            | $d_{hs}$ | 768    |
|                    | Hidden size of HAN output                | $d_{ha}$ | 100    |
|                    | Hidden size of initial node embedding     | $d_{hi}$ | 1,000  |
|                    | Hidden size of node embedding            | $d_h$    | 100    |
|                    | Hidden size of relation embedding        | $d_r$    | 100    |
| Multimodal training | Max sentence sample number                | $l_{m(max)}$ | 12    |
|                    | Training batch size                      | $b_{train}$ | 4     |
|                    | Test batch size                          | $b_{test}$ | 16    |
|                    | Weight decay                             | $w_d$    | $5 \times 10^{-5}$ |
|                    | Learning rate                            | $l_r$    | $1 \times 10^{-5}$ |
|                    | Gradient accumulate step                 | $step_g$ | 4      |
|                    | Optimizer                                |          | Adam   |
|                    | Scheduler                                |          | Linear |
|                    | Warmup rate                              | $r_{warmup}$ | 0.1 |
| Language model     | Max sentence sample number                | $l_{m(max)}$ | 12    |
|                    | Training batch size                      | $b_{train}$ | 4     |
|                    | Test batch size                          | $b_{test}$ | 16    |
|                    | Weight decay                             | $w_d$    | $5 \times 10^{-5}$ |
|                    | Learning rate                            | $l_r$    | $1 \times 10^{-5}$ |
|                    | Gradient accumulate step                 | $step_g$ | 4      |
|                    | Optimizer                                |          | Adam   |
|                    | Scheduler                                |          | Linear |
|                    | Warmup rate                              | $r_{warmup}$ | 0.1 |
| Graph model        | Training batch size                      | $b_{train}$ | 512   |
|                    | Test batch size                          | $b_{test}$ | 512   |
|                    | Weight decay                             | $w_d$    | $1 \times 10^{-8}$ |
|                    | Learning rate                            | $l_r$    | $1 \times 10^{-3}$ |
|                    | Optimizer                                |          | Adam   |
|                    | Scheduler                                |          | StepLR |
|                    | StepLR scheduler                         | $\gamma$ | 0.9    |