Pathway2Text: Dataset and Method for Biomedical Pathway Description Generation

Junwei Yang[1,2], **Zequn Liu**[2], **Ming Zhang**[2*], **Sheng Wang**[3*]
[1]School of Electronics Engineering and Computer Science, Peking University, Beijing, China
[2]School of Computer Science, Peking University, Beijing, China
[3]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA
yjwtheonly, zequnliu, mzhang_cs@pku.edu.cn
swang@cs.washington.edu

Abstract

Biomedical pathways have been extensively used to characterize the mechanism of complex diseases. One essential step in biomedical pathway analysis is to curate the description of a pathway based on its graph structure and node features. Neural text generation could be a plausible technique to circumvent the tedious manual curation. In this paper, we propose a new dataset Pathway2Text, which contains 2,367 pairs of biomedical pathways and textual descriptions. All pathway graphs are experimentally derived or manually curated. All textual descriptions are written by domain experts. We form this problem as a Graph2Text task and propose a novel graph-based text generation approach kNN-Graph2Text, which explicitly exploited descriptions of similar graphs to generate new descriptions. We observed substantial improvement of our method on both Graph2Text and the reverse task of Text2Graph. We further illustrated how our dataset can be used as a novel benchmark for biomedical named entity recognition. Collectively, we envision our method will become an important benchmark for evaluating Graph2Text methods and advance biomedical research for complex diseases.[1]

1 Introduction

Many complex diseases, such as cancer and neurodegenerative disorders, are driven by reactions among a combination of genes and metabolites instead of one single gene (Manolio et al., 2009). These reactions, which are formally referred to as pathways (Kanehisa et al., 2017; DS et al., 2020; Gillespie et al., 2022), are represented as a heterogeneous graph (**Figure** 1). Each node in this graph is a biomedical entity, such as gene, chemical or metabolite. Each edge is a specific biomedical reaction. Using natural language to describe this pathway graph is of great importance for scientific communication and further promotes applications in complex disease research (Whirl-Carrillo et al., 2012, 2021). To date, these descriptions are almost entirely curated manually by domain experts, thus substantially slowing down downstream biomedical applications (Naithani et al., 2019). Neural text generation has shown promising results in many applications (Bowman et al., 2016; Sutskever et al., 2014; Song et al., 2020; Brown et al., 2020; Raffel et al., 2020; Lewis et al., 2020). Among them, Graph-to-Text (Graph2Text) generation, such as AMR-to-Text (Song et al., 2018; Marcheggiani and Perez-Beltrachini, 2018; Fan and Gardent, 2020), and Knowledge-Graph-to-Text (Colas et al., 2021; Wang et al., 2021), is most similar to pathway description generation. Therefore, we hypothesize that neural text generation could also be a solution here. To fill in the gap, we first propose a novel biomedical pathway description dataset Pathway2Text, which contains 2,367 pairs of pathway and description. Each description is written by domain experts, describing the function and property of this pathway. In contrast to many other Graph2Text datasets (Banarescu et al., 2013; Colas et al., 2021) that use automatic approach to extract the graph from the text, pathways in our dataset are all experimentally measured or manually curated, presenting a high-quality structured data corresponding to the textual description. To the best of our knowledge, Pathway2Text is the first large-scale dataset studying the problem of biomedical pathway description generation.

One unique feature of our dataset is the rich textual information on each node in the graph. Specif-

*Corresponding author

[1]Our dataset is available at https://zenodo.org/record/6510039#.Ym9F15NBz0o. Our code is available at https://github.com/yjwtheonly/Pathway2Text.

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1441 - 1454
July 10-15, 2022

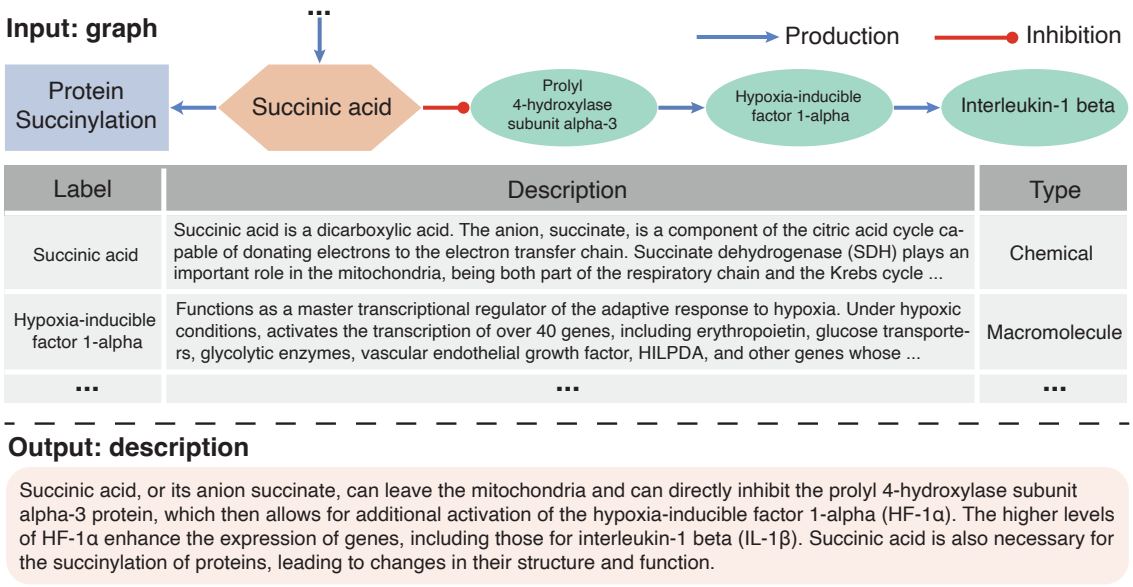

Figure 1: An example of a pathway and its description in our dataset. Each pathway is a heterogeneous graph containing different node types and edge types. Each node has three features: textual label, textual description and node type. For Graph2Text task, the input is the graph and the output is the graph description.

ically, each node is associated with a node type, a concise textual label and a detailed textual description. In contrast, many other Graph2Text datasets only have a short textual label or a fixed-size feature vector on each node (Belz et al., 2011; Banarescu et al., 2013; Gardent et al., 2017; Jin et al., 2020; Wang et al., 2021). We found that conventional graph neural network architectures were unable to fully exploited these rich node features, resulting in less accurate graph description generation. And the advantages of exploiting similar input data have been demonstrated in many related works (Baran et al., 2019; Khandelwal et al., 2020; Wang et al., 2022). We therefore propose kNN-Graph2Text, which explicitly incorporates descriptions of similar graphs into the definition generation process. In particular, our method first calculates a description-guided graph embedding and then finds similar graphs for a test graph based on these embeddings. After that, the new description is generated by jointly considering the description of neighbors and the graph structure using a multi-head attention framework (Vaswani et al., 2017).

We evaluated kNN-Graph2Text on our dataset and observed substantial improvement over conventional graph neural network architectures as well as methods that do not fully utilize the heterogeneous node features. We next demonstrated that our dataset can be used to study the reverse task of Text2Graph. In particular, we investigated how graph description can enhance the performance of link prediction and node classification, and obtained accuracy of 0.781 in link prediction and accuracy of 0.352 in node classification. Moreover, our dataset can be used as a novel benchmark for biomedical named entity recognition by extracting the ground truth entity types according to the annotated node types. Collectively, our dataset and our method present the first study in automatic biomedical pathway description generation. We envision Pathway2Text to be an important benchmark for general Graph2Text methods and facilitate downstream biomedical applications.

2 Pathway2Text Dataset

2.1 Data processing

Our dataset was synthesized from five biomedical databases: Reactome (Gillespie et al., 2022), KEGG (Kanehisa et al., 2017), Pathbank (DS et al., 2020), UniProt (Consortium, 2020) and ChEBI (Hastings et al., 2015). We collected biomedical pathways and their associated textual descriptions from Reactome, KEGG, and Pathbank, and aligned nodes in pathways to entities in UniProt and ChEBI for retrieving missing node descriptions. Specifically, the raw data was processed as follows:

Data format definition. We firstly modified SBGN (Le Novère et al., 2009) , a standard export format for biology graphical notation, to organize multiple components in pathway. We followed most of the original definitions in SBGN (e.g., reaction representing format), but we (1) omitted nodes that are not involved in any reaction (e.g., Compartment nodes), (2) reconstructed each Complex node

(a container for other nodes) into a tree structure, by adding additional edges between nodes and their container, (3) merged different nodes referring to the same entity into a single node for each pathway, and (4) rewrote the entire dataset into more readable JSON file (cf. Appendix B for more intuitive explanation of our data format).

Format translation. Reactome and Pathbank already provide SBGN files, making it straightforward to adapt pathways in both database into our data format. But KEGG only provides KGML (Kanehisa et al., 2017) file, a specific representation of KEGG pathways. So we applied additional modifications to translate KGML file into our data format: we (1) used Process node to represent a reaction instead of directly adding edges between substrates and products, (2) treated a Group of proteins acting on the same reaction as a Complex node, and (3) adjusted node types to match our definition. We refer readers to Appendix A for an illustration for these operations.

Node description gathering. Neither SBGN nor KGML file contains detailed node descriptions. SBGN file provides node label (a short text for display), and KGML file provides a KEGG identifier for each node. (1) For Pathbank database, each pathway is also recorded in PWML (DS et al., 2020) format, which contains textual node labels and descriptions. We therefore used node labels given by SBGN file to retrieve node descriptions from PWML file. (2) For Reactome database, each pathway is also stored in BioPAX (Demir et al., 2010) format. 49.2% nodes in BioPAX file have long descriptions while most of the others are only linked to identifiers in external biological entity databases. Among these databases, the Function attribute in UniProt and the Definition attribute in ChEBI are appropriate to be utilized as complements to node descriptions. So we aligned each node in SBGN file to node in corresponding BioPAX file using node label. And then extracted node descriptions from the union of BioPAX file, UniProt and ChEBI. (3) For KEGG database, each KEGG identifier indicates particular information (stored in a TXT file) of a specific entity. We parsed this file to pick entity name, textual Comment and external database identifiers. We used entity names as node labels, used Comments as node descriptions for entities having this attribute (3.7%), and used identifiers of UniProt and ChEBI to retrieve node descriptions for others.

2.2 Dataset description

After excluding duplicate pathways and pathways that do not have textual description, we finally obtained 2,367 pairs of pathway and description. An example is shown in **Figure** 1. Each textual description is a few sentences describing functions and structures of the pathway. The textual description has on average 129.5±101.4 words and 7.6±5.3 sentences. Each pathway can be viewed as a heterogeneous graph that contains different types of edges and nodes. There are 7 edge types and 7 node types in the entire dataset, where each pathway has on average 3.5±1.4 edge types and 4.5±1.4 node types. Each node type (e.g., chemical) has a large number of specific classes (e.g., succinic acid). Each class is associated with a concise textual label and a detailed textual description. The average length of the textual description is 114.8 words. We refer to the class description as the node description and the pathway description as the graph description throughout the paper. Each pathway has on average 61±52 nodes and 75±80 edges. In summary, there are four data fields for each pathway description pair: graph description, graph structure, node description and node label.

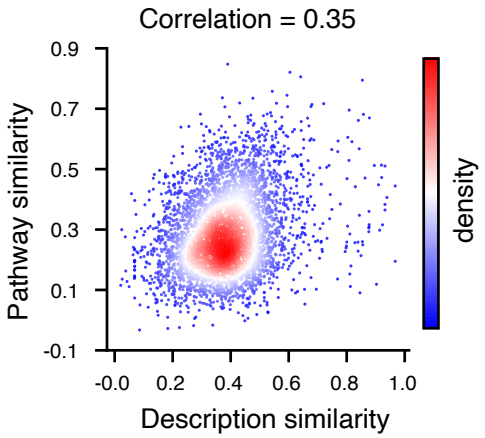

Figure 2: Scatter plot showing the consistency between graph-based representation similarity and description-based representation similarity. Each dot is a pair of graphs.

To examine the feasibility of conducting Graph2Text and Text2Graph tasks using our dataset, we examined the consistency between graph similarity and description similarity (**Figure 2**). We used GAT (Veličković et al., 2018) to embed each graph into a dense representation. We also obtained a dense representation for each graph description using BioBERT (Lee et al., 2020). For every two graphs, we calculated one similarity score based on their graph-based representations and an-

other similarity score based on their description-based representations. We observed a Pearson correlation 0.35 between these two similarity scores, reflecting a substantial consistency between these two similarity metrics. This indicates that graphs with similar structure tend to have similar textual descriptions, suggesting the possibility to generate textual description using the graph structure and vice versa.

3 Task Description

We aim to generate the textual description for a given biomedical pathway graph and generate the biomedical pathway graph from a given textual description. Let $\mathbf{D} = \{\mathbf{D}_{\mathcal{G}}, \mathbf{D}_{\mathcal{S}}\} = \{(G_i, S_i)\}_{i=1}^{N} \overset{dist}{\sim} \mathbb{P}(\mathcal{G}, \mathcal{S})$ be a dataset of paired pathway and its textual description. Each pathway is a directed graph $G = (V, E, F)$, where V represents the set of nodes, $E \subseteq V \times V$ represents the set of edges, and F represents node features. Since each pathway is a heterogeneous graph, we refer to pathway as graph in this paper.

One unique property of the graphs in our dataset is the rich node features $F = \{g, t, d\}$. In particular, each node v is associated with three features $\boldsymbol{g}_v$, t_v, and d_v. $\boldsymbol{g}_v \in \{0, 1\}^{n_c}$ is a one-hot vector representing the node type of v. $\boldsymbol{g}_v^i = 1$ if node v is type i. $t_v \triangleq \left\langle t_v^1, t_v^2, \ldots, t_v^{|t_v|} \right\rangle$ is the textual label of node v. $d_v \triangleq \left\langle d_v^1, d_v^2, \ldots, d_v^{|d_v|} \right\rangle$ is the textual description of node v. $t_v^i \in \mathrm{C}$ and $d_v^i \in \mathrm{C}$, where C is the vocabulary. In practice, the textual label is often a phrase and the textual definition is a few sentences. As a result, $|d_v|$ is often much larger than $|t_v|$. Each edge is associated with an edge type $r \in \mathrm{R}$, where R is the set of edge types in the dataset. Each graph description is a token sequence defined as $S \triangleq \langle S^1, S^2, \ldots, S^{|S|} \rangle$, where $S^i \in \mathrm{C}$.

We use an inductive learning framework in our experiment. The whole dataset $\mathbf{D}$ is randomly divided into $\mathbf{D}_{train} = \{(G_i, S_i)\}_{i=1}^{|\mathbf{D}_{train}|}$ and $\mathbf{D}_{test} = \{(G_i, S_i)\}_{i=|\mathbf{D}_{train}|+1}^{N}$. For each task, we train our model on $\mathbf{D}_{train}$ and evaluate its performance on $\mathbf{D}_{test}$. Graph G and textual description S are always observed for the training data. We define three tasks based on the unobserved information in the test data as follows:

Graph2Text. The input of this task is a graph G. All node features are observed on this graph. The output is the description text S for this graph.

Text2Graph link prediction. This task aims to predict missing links in a test graph. The inputs are graph description S, all node features F and a subset of edges $\{e\}$ in the graph G. For a test edge $e_{u,v} \in V \times V - \{e\}$, our goal is to classify $e_{u,v}$ into a specific edge type $r \in \mathrm{R}$.

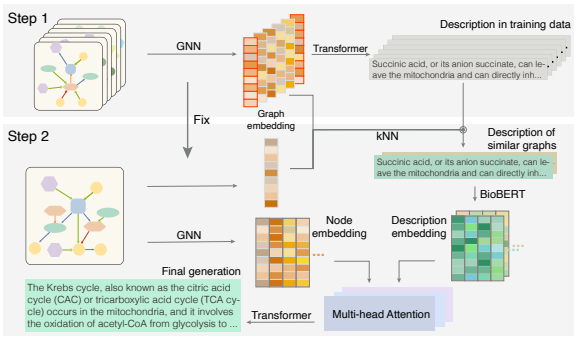

Figure 3: Flow chart of our two-step approach kNN-Graph2Text. In the first step, we learnt a representation for each graph by projecting graphs to descriptions. In the second step, we find similar graphs for a test graph and jointly use descriptions of similar graphs and node embeddings of the test graph to generate the final description.

Text2Graph node classification. This task aims to classify each test node into a specific node type in graph G. We split nodes in G into training nodes and test nodes. For training nodes, we observed all node features F, including textual label, textual description and node type, whereas none of these features is observed for the test node. We also observed the graph description S for G. Instead of predicting the node type, we aim at predicting the specific textual label, which is a more challenging task. We form this problem as a node classification task instead of textual generation.

4 Methods

4.1 Graph2Text

The overall framework of our method is shown in **Figure** 3. We propose a two-step approach. In the first step, we embed each graph into a dense representation through jointly considering its graph structure and node features. In the second step, we use the learnt graph embeddings to find similar graphs for each test graph and then leverage the description of these similar graphs to help the generation.

4.1.1 Description guided graph embedding

One unique property of our dataset is the rich textual features on each node. We hypothesize that unsupervised graph embedding methods might be unable to fully exploit these textual features. Therefore, we first use a supervised approach to obtain

graph embeddings. Since we don't have any class label for each graph, we treat the graph description as the pseudo label in the supervised learning framework to embed graphs.

In particular, we learn an encoder Enc that projects the graph G into a dense representation $\boldsymbol{h}_G$, and then a decoder Dec that maps this representation into the textual description S. The decoder will be discarded in the second step, while the encoder will be used to obtain the representation of an input graph.

Our encoder could be any existing graph neural network architectures (Kipf and Welling, 2017; Veličković et al., 2018; Xu et al., 2019). We first use a pretrained language model BioBERT to encode the textual label t_v and the description d_v of each node v into a dense vector $\boldsymbol{t}_v$ and a dense vector $\boldsymbol{d}_v$, and fuse them to get the initial node embedding for node v:

$$\boldsymbol{h}_v^0 = \text{RELU}([\boldsymbol{t}_v || \boldsymbol{d}_v]\mathbf{W}), \tag{1}$$

where $\mathbf{W}$ represents a trainable parameter matrix and $||$ is the concatenation operation.

We then propagate this embedding on the graph using a chosen graph neural network architecture, which learns representation of node v through iteratively updating it with neighbors' information $\boldsymbol{h}_{\mathcal{N}(v)}^l$ as:

$$\begin{aligned} \boldsymbol{h}_{\mathcal{N}(v)}^l &= \text{AGG}(\{(\boldsymbol{h}_u^{l-1}, e_{u,v}) | u \in \mathcal{N}(v)\}), \\ \boldsymbol{h}_v^l &= \text{UPDATE}(\boldsymbol{h}_v^{l-1}, \boldsymbol{h}_{\mathcal{N}(v)}^l), \end{aligned} \tag{2}$$

where $\mathcal{N}_v$ denotes the set of neighbors for v. AGG and UPDATE are the aggregation and the update function of the specific graph neural network architecture. We studied the performance of using GIN, GCN and GAT as the neural network architecture in our experiments.

After L iterations, the final embedding $\boldsymbol{h}_v^L$ can be used to represent the local subgraph comprising node v's L-hop neighbors. Next, for each node, we concatenate its node embeddings from all layers to fuse the information from different ranges of neighbors. We then calculate the graph-level representation by applying a READOUT function to the concatenated node embedding:

$$\begin{aligned} \boldsymbol{h}_v &= [\boldsymbol{h}_v^1 \| \boldsymbol{h}_v^2 \| \cdots \| \boldsymbol{h}_v^L]\mathbf{W}, \\ \boldsymbol{h}_G &= \text{READOUT}(\{\boldsymbol{h}_v\}_{v \in V}). \end{aligned} \tag{3}$$

Our decoder is a Transformer based on the pretrained BioBERT. It generates textual description conditioned on $\boldsymbol{h}_G$:

$$P(\hat{S}^i | \boldsymbol{h}_G) = \text{Dec}(\boldsymbol{h}_G, S^{1,\ldots,i-1}). \tag{4}$$

Finally, the decoder Dec and the encoder Enc are trained jointly using the following loss function:

$$\mathcal{L}_1 = -\frac{1}{|\mathbf{D}_{train}|} \sum_{(G,S) \in \mathbf{D}_{train}} \sum_{S^i \in S} \frac{\log P(S^i | \boldsymbol{h}_G)}{|S|}. \tag{5}$$

4.1.2 Exploiting descriptions of similar graphs in generation

The above encoder-decoder framework could already be used to generate the description for a given test graph. However, we observed that such generations were not of great quality in our experiment, partially due to the poor utilization of the node textual features. We thus propose to train a new decoder by leveraging the descriptions of similar graphs.

We first use $\boldsymbol{h}_{G_i}$ to find k similar graphs in the training data:

$$\begin{aligned} \text{dis}_{ij} &= \|\boldsymbol{h}_{G_i} - \boldsymbol{h}_{G_j}\|_{\text{F}}^2, \\ \bar{S}_i &= \mathop{\Vert}_{G_j \in k\text{NN}(G_i)} (S_j), \end{aligned} \tag{6}$$

where S_j is the description for k nearest graphs measured by dis_{ij}. We then embed neighbor's description $\bar{S}_i$ into a dense representation $\bar{\boldsymbol{s}}_i$ using BioBERT:

$$\begin{aligned} \langle \bar{\boldsymbol{s}}_i^j \rangle &= \text{BioBERT}(\bar{S}_i)\mathbf{W}, \\ \bar{\boldsymbol{s}}_i &= \text{Maxpooling}(\langle \bar{\boldsymbol{s}}_i^j \rangle). \end{aligned} \tag{7}$$

Next, we use multi-head attention framework to calculate a new dense representation $\boldsymbol{v}_s^a$ based on description embedding $\bar{\boldsymbol{s}}_i$ and $\langle \bar{\boldsymbol{s}}_i^j \rangle$, and a new dense representation $\boldsymbol{v}_g^a$ based on graph embedding $\boldsymbol{h}_G$ and $\{\boldsymbol{h}_v\}$as:

$$\begin{aligned} \text{s}^a(\boldsymbol{u}, \boldsymbol{v}_i, V) &= \frac{\exp(\mathcal{Q}^a(\boldsymbol{u})^{\text{T}} \mathcal{K}^a(\boldsymbol{v}_i))}{\sum_{\boldsymbol{v}_j \in V} \exp(\mathcal{Q}^a(\boldsymbol{u})^{\text{T}} \mathcal{K}^a(\boldsymbol{v}_j))}, \\ \text{Attention}^a(\boldsymbol{u}, V) &= \text{LeakyReLU}(\textstyle\sum_{\boldsymbol{v}_i \in V} \text{s}^a(\boldsymbol{u}, \boldsymbol{v}_i, V)\boldsymbol{v}_i), \\ \boldsymbol{v}_g^a &= \text{Attention}^a(\boldsymbol{h}_G, \{\boldsymbol{h}_v\}), \\ \boldsymbol{v}_s^a &= \text{Attention}^a(\bar{\boldsymbol{s}}_i, \langle \bar{\boldsymbol{s}}_i^j \rangle), \end{aligned} \tag{8}$$

where $a \in \{1, \ldots, A\}$ indicates the attention head number. $\mathcal{Q}^a$ is a projection function mapping a vector to the query space, which is defined as $\mathcal{Q}^a(\boldsymbol{v}) = \tanh(\boldsymbol{v}\mathbf{Q}^a)$, where $\mathbf{Q}^a$ represents a trainable parameter matrix. Similarly, we use $\mathcal{K}^a$ to map a vector to the key space.

Finally, we concatenate the new graph embedding $\boldsymbol{v}_g^a$ and new description embedding $\boldsymbol{v}_s^a$, and use a pretrained Transformer as the decoder to gen-

erate textual content:

$$\mathbf{V} = [\boldsymbol{v}_g^1||\cdots||\boldsymbol{v}_g^A||\boldsymbol{v}_s^1||\cdots||\boldsymbol{v}_s^A],$$
$$P(\hat{S}^i|\mathbf{V}) = \mathrm{Dec}(\mathbf{V}, S^{1,\ldots,i-1}). \quad (9)$$

Since we didn't use the position embedding in the input of the Transformer encoder, it implicitly performs cross attention between graph and description. The loss function is finally defined as:

$$\mathcal{L}_2 = -\frac{1}{|\mathbf{D}_{train}|}\sum_{(D,S)\in\mathbf{D}_{train}}\sum_{S^i\in S}\frac{\log P(S^i|\mathbf{V})}{|S|}. \quad (10)$$

4.2 Text2Graph

For Text2Graph, we studied link prediction and node classification.

4.2.1 Link prediction

To predict the edge type between node u and node v on graph G, we used the node embedding $\boldsymbol{h}_u$, node embedding $\boldsymbol{h}_v$ and the graph description S as the input features. We first define the edge feature $\boldsymbol{w}_{u,v}$ and the graph description feature $\langle \boldsymbol{s}_i^j \rangle$ as:

$$\langle \boldsymbol{s}_i^j \rangle = \text{BioBERT}(S_i)\mathbf{W},$$
$$\boldsymbol{w}_{u,v} = [\boldsymbol{h}_u||\boldsymbol{h}_v]. \quad (11)$$

Then we use the same attention mechanism as in Equation. 8 to obtain a new embedding $\boldsymbol{h}$ from these two features and define the predicted distribution $P(\hat{r}_{u,v}|e_{u,v})$ for edge type r as:

$$\boldsymbol{h} = \text{Attention}(\boldsymbol{w}_{u,v}, \langle \boldsymbol{s}_i^j \rangle),$$
$$P(\hat{r}_{u,v}|S) = \text{softmax}(\text{MLP}([\boldsymbol{h}_u||\boldsymbol{h}_v||\boldsymbol{h}])). \quad (12)$$

Here, MLP is a multi-layer perceptron. The final training loss is defined as:

$$\mathcal{L}_3 = -\frac{1}{|\mathbf{D}_{train}|}\sum_{(G,S)\in\mathbf{D}_{train}}\sum_{e_{u,v}}\frac{P(r_{u,v}|S)}{|\{e_{u,v}\}|}. \quad (13)$$

4.2.2 Node classification

To classify a test node v, we applied a similar attention mechanism on its node embedding $\boldsymbol{h}_v$ and graph description feature $\langle \boldsymbol{s}_i^j \rangle$ as:

$$\langle \boldsymbol{s}_i^j \rangle = \text{BioBERT}(S_i)\mathbf{W},$$
$$\boldsymbol{h} = \text{Attention}(\boldsymbol{h}_v, \langle \boldsymbol{s}_i^j \rangle). \quad (14)$$

We then define the predicted label distribution and loss function accordingly as:

$$P(\hat{t}_v|S) = \text{softmax}(\text{MLP}([\boldsymbol{h}_v||\boldsymbol{h}])),$$
$$\mathcal{L}_4 = -\frac{1}{|\mathbf{D}_{train}|}\sum_{(G,S)\in\mathbf{D}_{train}}\sum_{v}\frac{P(t_v|S)}{|\{v\}|}. \quad (15)$$

5 Results

5.1 Experimental setup

We exclude any pathway that is a subgraph of another pathway in all experiments to avoid data leakage. For Graph2Text, we randomly split the graph description pairs into 75% training pairs and 25% test pairs. We used a fixed Transformer encoder in BioBERT and initialized the GNN with xavier initialization. We used a learning rate 5e-5. We found that this method performed better than using a fixed Transformer and warming GNN before the training. We used GAT (Veličković et al., 2018), GCN (Kipf and Welling, 2017) and GIN (Xu et al., 2019) as different graph encoders. The hidden state embedding dimension was set to 128 for GAT and 512 for others. The number of heads of GAT was set as 4. AGG and UPDATE functions were implemented according to the original papers. Global mean pooling was used as the READOUT function. Since Transformer can hardly generate more than 512 tokens, we calculated the loss functions and evaluated the generation only on the first 3 sentences, which have an average token length 69±23 (maximum token length is 471). However, the entire text was used as the input in all tasks through the attention mechanism, and we set the attention head number $A = 128$. We set k to 1 in the kNN framework. We focused on the 1,173 pathway from Pathbank (DS et al., 2020) in our experiments.

For Text2Graph node classification, we randomly split the graph and description pairs into 75% training pairs and 25% test pairs. We sampled 10% nodes as the test node in each graph. In Text2Graph link prediction task, we varied the proportion of the test set (10%, 30%, 50%, 70%, 90%). We sampled 40% edges for each graph and the same number of edges from the complementary graph as the test edge. In link prediction and node classification, we only used GAT since it obtained the best performance in Graph2Text. We set the learning rate to 5e-4. We used Adam optimizer for all optimizations.

In Graph2Text task, we compared our methods to supervised graph neural network which jointly trains a graph neural network and a transformer. We denote them as **GNN (des.)**, **GNN (label)**,**GNN (des. + label)** and **GNN(structure only)** based on the node features used. In particular, **GNN (des.)** uses textual description as node feature. **GNN (label)** uses textual label as the node feature. **GNN (des. + label)** uses both textual label

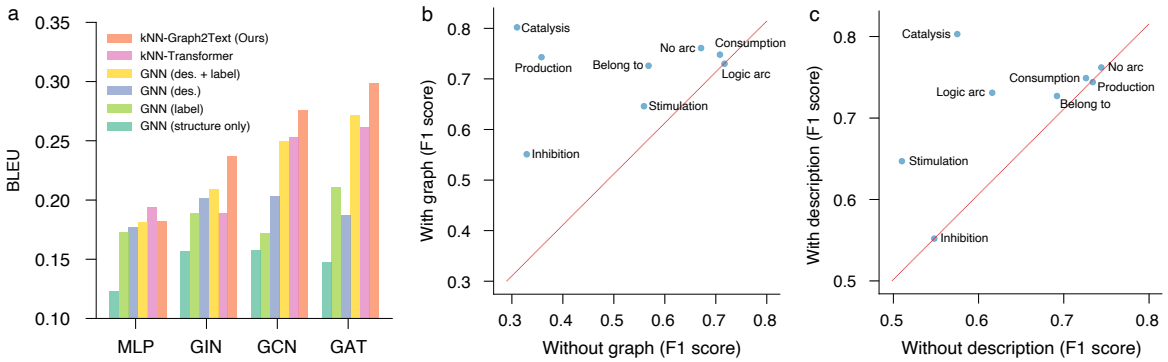

Figure 4: **Performance of our method on Graph2Text and Text2Graph link prediction. a**, Bar plot comparing our method and baselines using different graph neural network architectures on Graph2Text. **b**, Scatter plot comparing the F1 score of using the graph structure to the F1 score of without using the graph structure. Each dot is one edge type. **c**, Scatter plot comparing the F1 score of using the graph description to the F1 score of without using the graph description. Each dot is one edge type.

Method	BLEU1	BLEU2	BLEU3	METEOR	NIST	ROUGE-L
GNN (structure only)	14.3	2.2	0.9	12.1	0.8	19.4
GNN (des.)	18.7	2.5	0.9	11.9	1.1	16.6
GNN (label)	21.4	4.2	1.3	13.2	1.2	17.1
GNN (des. + label)	27.1	11.9	10.8	20.5	1.9	23.9
kNN-Transformer	26.8	12.3	10.6	20.4	1.9	24.3
kNN-Graph2Text (Ours)	**29.6**	**13.8**	**11.4**	**23.0**	**2.2**	**24.4**

Table 1: Comparison on Graph2Text using different metrics.

and description as the node feature. We also compared to a k**NN-Transformer** model which trained a transformer using descriptions of similar graphs to the final description. Different GNN architectures are used to identify nearest neighbors in kNN based on the graph information.

5.2 Graph2Text

We sought to evaluate the performance of our method on the task of Graph2Text (**Figure 4a**, **Table 1**). Overall, we found that our method achieves the best performance on all metrics (0.296 BLEU-1 score, 0.230 METEOR, 2.2 NIST, and 0.244 ROGUE-L), demonstrating the effectiveness of jointly modeling graph structure, node description and node label. We first compared our method to graph neural network, which performed the first step of our framework and used concatenated node embeddings instead of single graph embedding as the input to Transformer. We observed substantial improvement over it on all three kinds of graph neural networks, indicating the importance of retraining using descriptions of similar graphs. We also observed that our method was better than kNN-Transformer, reflecting how our description-guided graph embeddings enhance the description generation.

To further understand the importance of each type of node feature, we evaluate the variants that only consider node description or node textual label (**Figure 4a**). We found that the performance of both variants dropped substantially, demonstrating the importance of both node textual label and node description. We further observed that the improvement of our method was consistent when using other graph neural network architectures, including GIN and GCN, demonstrating the robustness of our method. When replacing GAT to a multi-layer perception that cannot model the graph structure, the BLEU score of our method dropped substantially from 0.296 to 0.187, again confirming the necessity of considering the graph structure in this task.

5.3 Text2Graph

We next investigated the performance on the task of Text2Graph. Here, we studied two classic graph prediction tasks: link prediction and node classification. We summarized the performance of link prediction in **Figure 5a**. We obtained an average of 0.781 accuracy score across 8 different edge types, demonstrating an accurate prediction of the graph structure using the graph description. We further examined the effect of using the graph description in **Figure 4c** and observed that all 8 edge types had better F1 score when the graph description was used. We observed the same improvement of using the graph description when evaluated using the accuracy. We also performed the ablation study for the graph structure and observed similar improvement **Figure 4b**. These results collectively confirm that our method can generate the graph structure based on the graph description, offering biologists novel insights in pathway analysis.

We then studied the performance of node classification. We considered three most frequent node types in our dataset: macromolecule, multimer

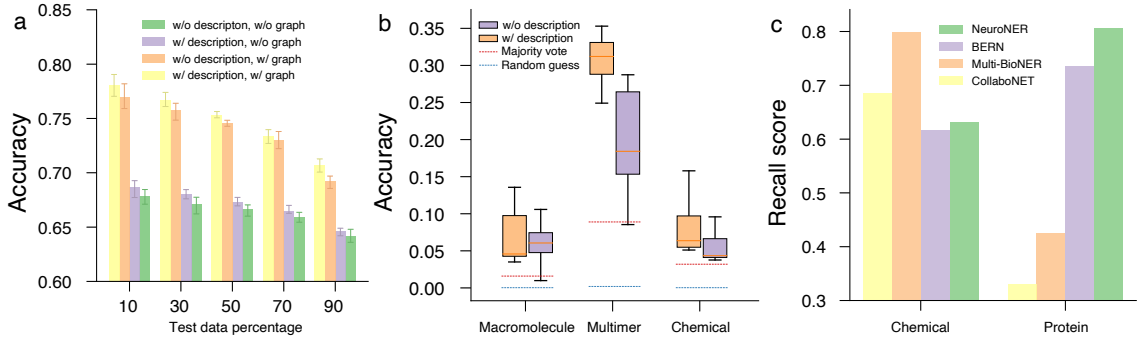

Figure 5: **Performance on Text2Graph link prediction, node classification and named entity recognition. a**, Bar plot showing the ablation studies on using the graph description and using the graph structure on link prediction. **b**, Box plot showing the comparison between using the graph description and without using the graph description on node classification. **c**, Bar plot showing the performance of named entity recognition on chemical and protein on our dataset.

and chemical. For each node type, we formed the node classification task as a multi-class classification problem, where each test node is classified into a specific class defined by the textual label. We noticed that each node type has a large number of classes. Therefore, we first evaluated two naive baselines: random guess and majority vote. Random guess obtained 0.0009 average accuracy, while majority vote obtained 0.046 average accuracy, suggesting a challenging classification task. Our method obtained a desirable classification performance, which was substantially higher than the performance of the variant that does not consider the graph description (**Figure 5b**). The improvement of using graph description on both node classification and link prediction further confirm that our dataset could be a promising benchmark for Text2Graph task.

6 Application to Named Entity Recognition

Named entity recognition (NER) is essential in detecting chemicals, genes, and diseases from biomedical text (Leaman et al., 2016; Luo et al., 2018; Kim et al., 2019; Yoon et al., 2019), and further facilitating downstream bioNLP applications, such as relation extraction(Xing et al., 2020). A major bottleneck in NER is the lack of curated benchmarks since such curation often requires substantial domain expertise. Our dataset Path2wayText can be used as a novel curated benchmark for NER.

Specifically, we used the graph description as the sentences that one wants to perform NER. We then obtained the ground truth entity type of phrases in these sentences according to their curated node types in the graph. Since the graphs, including all node types, are curated by domain experts, such node types can be used as the ground truth entity types for NER. Here, we focused on two most frequent entity types in our dataset: protein and chemical. We noticed that some phrases in the graph description sentences might also be a protein or chemical, even though they were not curated in the graph. We excluded such phrases in the evaluation in order to maintain the quality of our NER benchmark.

To this end, we obtained the graph-based curation of 8,779 protein entities and 1,621 chemical entities, offering a good complementary to existing biomedical NER datasets (Kim et al., 2003; Smith et al., 2008; Doğan et al., 2014; Krallinger et al., 2015; Li et al., 2016; Wei et al., 2018). To further investigate the performance of our novel NER datasets, we tested a few state-of-the-art biomedical NER methods, including BERN (Kim et al., 2019), CollaboNet (Yoon et al., 2019), Multi-BioNER (Wang et al., 2019), and NeuroNER (Dernoncourt et al., 2017). We observed that NeuroNER obtained the best performance on protein and Multi-BioNER achieved the best performance on Chemical (**Figure 5c**). Moreover, existing approaches only consider the graph description sentences when labelling entity types. In addition to graph description, our dataset also contains the corresponding graph structure, which has been shown to be critical in graph description generation in our experiments. Therefore, we hypothesize that graph structure might be also helpful in NER, and envision our dataset to be an important resource for benchmarking graph-based NER methods (Radford et al., 2015; Rijhwani et al., 2020; He et al., 2020; Nie et al., 2021).

7 Related Work

Graph2Text, which aims at generating a textual description for a structured graph, has attracted attentions in different applications. Existing Graph2Text datasets aims to generate text from RDF data (Gardent et al., 2017), knowledge graph (Koncel-Kedziorski et al., 2019; Jin et al., 2020; Cheng et al., 2020; Colas et al., 2021; Wang et al., 2021), street view map (Schumann and Riezler, 2021), Abstract Meaning Representation (AMR) (Banarescu et al., 2013; Marcheggiani and Perez-Beltrachini, 2018; Song et al., 2018; Ribeiro et al., 2019; Zhu et al., 2019; Hajdik et al., 2019; Damonte and Cohen, 2019; Mager et al., 2020; Zhang et al., 2020; Zhao et al., 2020; Fan and Gardent, 2020; Wang et al., 2020), terminology ontology (Liu et al., 2021) and graph-transduction grammars (Belz et al., 2011; Mille et al., 2019, 2020). Our dataset is the first Graph2Text dataset that focuses on biomedical pathway generation. In addition, our dataset has more complicated node features than many existing Graph2Text datasets, where each node in our dataset has a node type, a concise textual label and a detailed textual description.

Text2Graph can be viewed as an information extraction task, which aims at mining structured knowledge from free text. The datasets that are more relevant to our task could be generating a knowledge graph from long document (Kertkeidkachorn and Ichise, 2017; Bosselut et al., 2019; Kannan et al., 2020; Wu et al., 2020). Many of these existing datasets use automatic annotation to extract the graph information from corpus (Kertkeidkachorn and Ichise, 2017; Bosselut et al., 2019), which might introduce bias from the extraction method. In contrast, graphs in our dataset are either experimentally derived or manually curated, presenting a high-quality complementary to existing Text2Graph datasets.

8 Conclusion and Future work

We have presented a novel dataset Pathway2Text for biomedical pathway description generation. Our dataset contains 2,367 pairs of curated pathway and its associated description. To generated description for biomedical pathways, we have proposed a kNN-Graph2Text approach, which utilizes neighbor's description to enhance the text generation. We have extensively evaluated our method and observed substantial improvement in comparison to conventional graph neural network architectures. Furthermore, we have investigated the reverse task of Text2Graph and illustrated how our dataset can serve as a novel benchmark for biomedical NER.

In addition to Graph2Text, Text2Graph and NER, our dataset can also be used to investigate other important applications. For example, our dataset can be used as a relation extraction benchmark by regarding graph descriptions as sentences and graph edge types as the ground truth relation type. We can also use our dataset to study other graph-based tasks, such as generating node description given the graph structure and the graph description. Another interesting application is to identify the importance of each node in the graph, which has important applications in recommender system and social media. The order of mentions of each node in the graph description can be used to evaluate the node importance since the graph description often starts from the most important node.

From a methodological perspective, we plan to develop semi-supervised approaches to leverage many other biomedical pathways that currently do not have curated description. For example, we can train a Graph Transformer (Cai and Lam, 2020) on these unlabelled pathways and then fine-tune the model on pathways with graph description. We also want to explore other geometric embedding methods, such as hyperbolic embedding (Cvetkovski and Crovella, 2009) and spherical embedding (Meng et al., 2019, 2020), since biomedical pathways often form a hierarchical structure.

More importantly, our dataset could also open up new venues in biomedical research. Any computational biology tools that utilize biomedical pathways as features in their pipeline can exploit the graph description as additional features. For biomedical pathways that do not have the corresponding description, one can use the description generated by our kNN-Graph2Text as the feature. We envision this will substantially advance a wide range of biomedical research that involves pathway analysis, and our dataset will introduce other new text generation tools developed in the NLP community to broader audience in biomedicine.

Acknowledgement

This paper is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101902 as well as the National Natural Science Foundation of China (NSFC Grant No. 62106008 and No. 62006004).

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*.

Yael Baran, Akhiad Bercovich, Arnau Sebe-Pedros, Yaniv Lubling, Amir Giladi, Elad Chomsky, Zohar Meir, Michael Hoichman, Aviezer Lifshitz, and Amos Tanay. 2019. Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome Biology*, 20(1):206.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 217–226.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *ACL*, pages 4762–4779.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. pages 10–21.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *AAAI*, pages 7464–7471.

Liying Cheng, Dekun Wu, Lidong Bing, Yan Zhang, Zhanming Jie, Wei Lu, and Luo Si. 2020. ENT-DESC: Entity description generation by exploring knowledge graph. In *EMNLP 2020*, pages 1187–1197.

Anthony Colas, Ali Sadeghian, Yue Wang, and Daisy Zhe Wang. 2021. Eventnarrative: A large-scale event-centric dataset for knowledge graph-to-text generation. *CoRR*, abs/2111.00276.

The UniProt Consortium. 2020. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489.

Andrej Cvetkovski and Mark Crovella. 2009. Hyperbolic embedding and routing for dynamic graphs. In *INFOCOM*, pages 1647–1655.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for amr-to-text generation. In *NAACL-HLT 2019, June 2-7, 2019, Volume 1*, pages 3649–3658.

Emek Demir, Michael P. Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'Eustachio, Carl Schaefer, Joanne Luciano, Frank Schacherer, Irma Martinez-Flores, Zhenjun Hu, Veronica Jimenez-Jacinto, Geeta Joshi-Tope, Kumaran Kandasamy, Alejandra C. Lopez-Fuentes, Huaiyu Mi, Elgar Pichler, Igor Rodchenkov, Andrea Splendiani, Sasha Tkachev, Jeremy Zucker, Gopal Gopinath, Harsha Rajasimha, Ranjani Ramakrishnan, Imran Shah, Mustafa Syed, Nadia Anwar, Özgün Babur, Michael Blinov, Erik Brauner, Dan Corwin, Sylva Donaldson, Frank Gibbons, Robert Goldberg, Peter Hornbeck, Augustin Luna, Peter Murray-Rust, Eric Neumann, Oliver Ruebenacker, Matthias Samwald, Martijn van Iersel, Sarala Wimalaratne, Keith Allen, Burk Braun, Michelle Whirl-Carrillo, Kei-Hoi Cheung, Kam Dahlquist, Andrew Finney, Marc Gillespie, Elizabeth Glass, Li Gong, Robin Haw, Michael Honig, Olivier Hubaut, David Kane, Shiva Krupa, Martina Kutmon, Julie Leonard, Debbie Marks, David Merberg, Victoria Petri, Alex Pico, Dean Ravenscroft, Liya Ren, Nigam Shah, Margot Sunshine, Rebecca Tang, Ryan Whaley, Stan Letovksy, Kenneth H. Buetow, Andrey Rzhetsky, Vincent Schachter, Bruno S. Sobral, Ugur Dogrusoz, Shannon McWeeney, Mirit Aladjem, Ewan Birney, Julio Collado-Vides, Susumu Goto, Michael Hucka, Nicolas Le Novère, Natalia Maltsev, Akhilesh Pandey, Paul Thomas, Edgar Wingender, Peter D. Karp, Chris Sander, and Gary D. Bader. 2010. The biopax community standard for pathway data sharing. *Nature Biotechnology*, 28(9):935–942.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *J Biomed Inform*, 47:1–10.

Wishart DS, Li C, Marcu A, Badran H, Pon A, Budinski Z, Patron J, Lipton D, Cao X, Oler E, Li K, Paccoud M, Hong C, Guo AC, Chan C, Wei W, and Ramirez-Gaona M. 2020. Pathbank: a comprehensive pathway database for model organisms. In *Nucleic Acids Res*.

Angela Fan and Claire Gardent. 2020. Multilingual AMR-to-text generation. In *EMNLP2020*, pages 2889–2901.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, et al. 2022. The reactome pathway knowledgebase 2022. *Nucleic acids research*, 50(D1):D687–D692.

Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. 2019. Neural text generation from rich semantic representations. In *NAACL-HLT 2019, June 2-7, 2019, Volume 1*, pages 2259–2266.

Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. 2015. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res*, 44(D1):D1214–9.

Qizhen He, Liang Wu, Yida Yin, and Heming Cai. 2020. Knowledge-graph augmented word representations for named entity recognition. In *EAAI*, pages 7919–7926.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409.

Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. 2017. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic acids research*, 45(D1):D353–D361.

Amar Viswanathan Kannan, Dmitriy Fradkin, Ioannis Akrotirianakis, Tugba Kulahcioglu, Arquimedes Canedo, Aditi Roy, Shih-Yuan Yu, Arnav V. Malawade, and Mohammad Abdullah Al Faruque. 2020. Multimodal knowledge graph for deep learning papers and code. In *CIKM*, pages 3417–3420.

Natthawut Kertkeidkachorn and Ryutaro Ichise. 2017. T2KG: an end-to-end system for creating knowledge graph from unstructured text. In *AAAI*, volume WS-17 of *AAAI Workshops*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *ICLR*.

Donghyeon Kim, Jinhyuk Lee, Chan Ho So, Hwisang Jeon, Minbyul Jeong, Yonghwa Choi, Wonjin Yoon, Mujeen Sung, and Jaewoo Kang. 2019. A neural named entity recognition and multi-type normalization tool for biomedical text mining. *IEEE Access*, 7:73729–73740.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *NAACL-HLT 2019, June 2-7, 2019, Volume 1*, pages 2284–2293.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2.

Nicolas Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I. Aladjem, Sarala M. Wimalaratne, Frank T. Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villéger, Sarah E. Boyd, Laurence Calzone, Melanie Courtot, Ugur Dogrusoz, Tom C. Freeman, Akira Funahashi, Samik Ghosh, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, Esther Schmidt, Steven Watterson, Guanming Wu, Igor Goryanin, Douglas B. Kell, Chris Sander, Herbert Sauro, Jacky L. Snoep, Kurt Kohn, and Hiroaki Kitano. 2009. The Systems Biology Graphical Notation. *Nature Biotechnology*, 27(8):735–741.

Robert Leaman, Chih-Hsuan Wei, Cherry Zou, and Zhiyong Lu. 2016. Mining chemical patents with an ensemble of open systems. *Database J. Biol. Databases Curation*, 2016.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Zequn Liu, Shukai Wang, Yiyang Gu, Ruiyi Zhang, Ming Zhang, and Sheng Wang. 2021. Graphine: A dataset for graph-aware terminology definition generation. In *EMNLP 2021, 7-11 November, 2021*, pages 3453–3463.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. 2018. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinform.*, 34(8):1381–1388.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852.

Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, et al. 2009. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation, November 5-8, 2018*, pages 1–9.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance M. Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8206–8215.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD*, pages 1908–1917.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *MSR 2019*, pages 1–17.

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR'20): Overview and evaluation results. In *MSR 2020*, pages 1–20.

Sushma Naithani, Parul Gupta, Justin Preece, Priyanka Garg, Valerie Fraser, Lillian K Padgitt-Cobb, Matthew Martin, Kelly Vining, and Pankaj Jaiswal. 2019. Involving community in genes and pathway curation. *Database*, 2019.

Binling Nie, Ruixue Ding, Pengjun Xie, Fei Huang, Chen Qian, and Luo Si. 2021. Knowledge-aware named entity recognition with alleviating heterogeneity. In *AAAI*, pages 13595–13603.

Will Radford, Xavier Carreras, and James Henderson. 2015. Named entity recognition with document-specific KB tag gazetteers. In *EMNLP*, pages 512–517.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing amr-to-text generation with dual graph representations. In *EMNLP-IJCNLP 2019, November 3-7, 2019*, pages 3181–3192.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime G. Carbonell. 2020. Soft gazetteers for low-resource named entity recognition. In *ACL*, pages 8118–8123.

Raphael Schumann and Stefan Riezler. 2021. Generating landmark navigation instructions from maps as a graph-to-text problem. In *ACL/IJCNLP 2021, Volume 1, Virtual Event, August 1-6, 2021*, pages 489–502.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):1–19.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *ACL 2018, July 15-20, 2018, Volume 1*, pages 1616–1626.

Yiping Song, Zequn Liu, Wei Bi, Rui Yan, and Ming Zhang. 2020. Learning to customize model structures for few-shot dialogue generation tasks. In *ACL*, pages 5832–5841.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Luyu Wang, Yujia Li, Özlem Aslan, and Oriol Vinyals. 2021. Wikigraphs: A wikipedia text - knowledge graph paired dataset. *CoRR*, abs/2107.09556.

Mu-Chun Wang, Zixuan Liu, and Sheng Wang. 2022. Textomics: A dataset for genomics data summary generation. In *ACL*.

Tianming Wang, Xiaojun Wan, and Shaowei Yao. 2020. Better amr-to-text generation with graph structure reconstruction. In *IJCAI-20, Organization*, pages 3919–3925.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2019. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Chih-Hsuan Wei, Lon Phan, Juliana Feltz, Rama Maiti, Tim Hefferon, and Zhiyong Lu. 2018. tmvar 2.0: integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. *Bioinformatics*, 34(1):80–87.

Michelle Whirl-Carrillo, Rachel Huddart, Li Gong, Katrin Sangkuhl, Caroline F Thorn, Ryan Whaley, and Teri E Klein. 2021. An evidence-based framework for evaluating pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 110(3):563–572.

Michelle Whirl-Carrillo, Ellen M McDonagh, JM Hebert, Li Gong, K Sangkuhl, CF Thorn, Russ B Altman, and Teri E Klein. 2012. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 92(4):414–417.

Tianxing Wu, Haofen Wang, Cheng Li, Guilin Qi, Xing Niu, Meng Wang, Lin Li, and Chaomin Shi. 2020. Knowledge graph construction from multiple online encyclopedias. *World Wide Web*, 23(5):2671–2698.

Rui Xing, Jie Luo, and Tengwei Song. 2020. Biorel: towards large-scale biomedical relation extraction. *BMC Bioinform.*, 21-S(16):543.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *ICLR*.

Wonjin Yoon, Chan Ho So, Jinhyuk Lee, and Jaewoo Kang. 2019. Collabonet: collaboration of deep neural networks for biomedical named entity recognition. *BMC bioinformatics*, 20(10):55–65.

Yan Zhang, Zhijiang Guo, Zhiyang Teng, Wei Lu, Shay B. Cohen, Zuozhu Liu, and Lidong Bing. 2020. Lightweight, dynamic graph convolutional networks for AMR-to-text generation. In *EMNLP2020*, pages 2162–2172.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 732–741.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better amr-to-text generation. In *EMNLP-IJCNLP 2019, November 3-7, 2019*, pages 5458–5467.

A KGML Translation

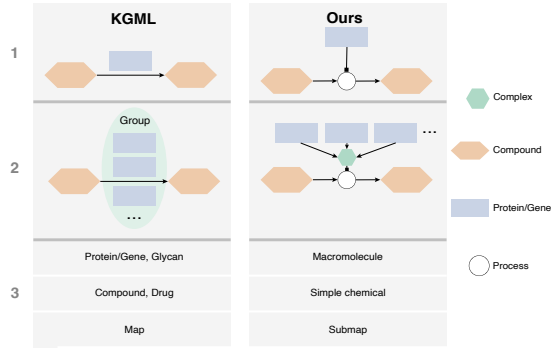

Figure 6: An illustration of KGML translation mentioned in Section 2.1. The first and second operations aim to unify the expression of reaction. The third operation aims to eliminate inconsistencies between node types. The Orthology nodes in KGML file are omitted during this translation.

B Data Format

Our dataset is stored in a JSON file. And the hierarchy structure is organized as follows:

```
{
    Graph identifier: {
        "Name": ,
        "Graph_description": ,
        "Node_dict": {
            Node identifier: {
                "type": ,
                "label": ,
                "description": .
            },
            ...
        },
        "Arc_list": [
            {
              "arc_sourse": ,
              "arc_target": ,
              "arc_type": .
            },
            ...
        ]
    },
    ...
}.
```

The node types include Submap, Macromolecule, Process, Complex, Multimer, Simple Chemical and Others. The Others is the union of several types occurring only in a single database (e.g., Unspecified Entity, Association in Reactome and Transport in Pathbank). Nodes in this type account for 7% over the whole dataset. The edge types include Catalysis, Consumption, Stimulation, Inhibition, Production, Logic Arc and Belong To, where the Belong To represents edges that were added for Complex node reconstruction mentioned in Section 2.1.