

Graphine: A Dataset for Graph-aware Terminology Definition Generation

Zequn Liu¹, Shukai Wang¹, Yiyang Gu¹, Ruiyi Zhang¹, Ming Zhang^{1*}, Sheng Wang^{2*}

¹Department of Computer Science, School of EECS, Peking University, Beijing, China

²Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA

zequnliu, shukaiwang, yiyangu, zhangruiyi, mzhang_cs@pku.edu.cn

swang@cs.washington.edu

Abstract

Precisely defining the terminology is the first step in scientific communication. Developing neural text generation models for definition generation can circumvent the labor-intensive curation, further accelerating scientific discovery. Unfortunately, the lack of large-scale terminology definition dataset hinders the process toward definition generation. In this paper, we present a large-scale terminology definition dataset Graphine covering 2,010,648 terminology definition pairs, spanning 227 biomedical subdisciplines. Terminologies in each subdiscipline further form a directed acyclic graph, opening up new avenues for developing graph-aware text generation models. We then proposed a novel graph-aware definition generation model Graphex that integrates transformer with graph neural network. Our model outperforms existing text generation models by exploiting the graph structure of terminologies. We further demonstrated how Graphine can be used to evaluate pretrained language models, compare graph representation learning methods and predict sentence granularity. We envision Graphine to be a unique resource for definition generation and many other NLP tasks in biomedicine.¹

1 Introduction

Obtaining the definition is the first step toward understanding a new terminology. The lack of precise terminology definition poses great challenges in scientific communication and collaboration (Oke, 2006; Cimino et al., 1994), which further hinders new discovery. This problem becomes even more severe in emerging research topics (Baig, 2020; Baines et al., 2020), such as COVID-19, where curated definitions could be imprecise and do not scale to rapidly proposed terminologies.

*Corresponding author

¹Our Dataset is available at <https://zenodo.org/record/5320310#.YSxHgI77Q2w>. Our code is available at <https://github.com/zequnl/Graphex>

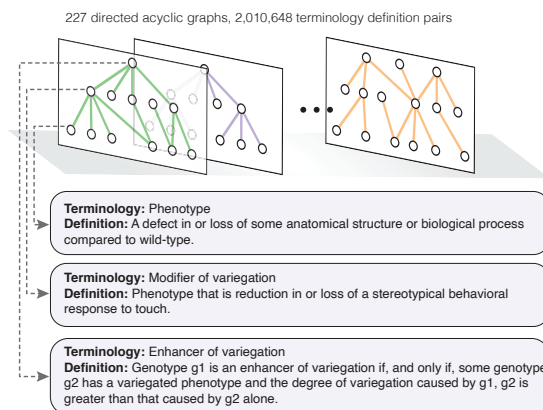


Figure 1: Graphine dataset contains 2,010,648 terminology definition pairs organized in 227 directed acyclic graphs. Each node in the graph is associated with a terminology and its definition. Terminologies are organized from coarse-grained ones to fine-grained ones in each graph.

Neural text generation (Bowman et al., 2016; Vaswani et al., 2017; Sutskever et al., 2014; Song et al., 2020b) could be a plausible solution to this problem by generating definition text based on the terminology text. Encouraging results by neural text generation have been observed on related tasks, such as paraphrase generation (Li et al., 2020), description generation (Cheng et al., 2020), synonym generation (Gupta et al., 2015) and data augmentation (Malandrakis et al., 2019). However, it remains unclear how to generate definition, which comprises concise text in the input space (i.e., terminology) and longer text in the output space (i.e., definition). Moreover, the absence of large-scale terminology definition datasets impedes the progress towards developing definition generation models.

Despite these challenges, scientific terminologies often form a directed acyclic graph (DAG), which could be helpful in definition generation. Each DAG organizes related terminologies from general ones to specific ones with different granu-

larity levels (**Figure 1**). These DAGs have proved to be useful in assisting disease, cell type and function classification (Wang et al., 2020b; Song et al., 2020a; Wang et al., 2015) by exploiting the principle that nearby terms on the graph are semantically similar (Altshuler et al., 2000). Likewise, terminologies that are closer on this DAG should acquire similar definitions. Moreover, placing a new terminology in an existing DAG requires considerably less expert efforts than curating the definition, further motivating us to generate the definition using the DAG.

In this paper, we collectively advance definition generation in the biomedical domain through introducing a terminology definition dataset Graphine and a novel graph-aware text generation model Graphex. Graphine encompasses 2,010,648 terminology definition pairs encapsulated in 227 DAGs. These DAGs are collected from three major biomedical ontology databases (Smith et al., 2007; Noy et al., 2009; Jupp et al., 2015). All definitions are curated by domain experts. Our graph-aware text generation model Graphex utilizes the graph structure to assist definition generation based on the observation that nearby terminologies exhibit semantically similar definitions.

Our human and automatic evaluations demonstrate the substantial improvement of our method on definition generation in comparison to existing text generation methods that do not consider the graph structure. In addition to definition generation, we illustrate how Graphine opens up new avenues for investigating other tasks, including domain-specific language model pretraining, graph representation learning and a novel task of sentence granularity prediction. Finally, we present case studies of a failed generation by our method, pinpointing directions for future improvement. To the best of our knowledge, Graphine and Graphex build up the first large-scale benchmark for terminology definition generation, and can be broadly applied to a variety of tasks.

2 Graphine Dataset

2.1 Data collection and statistics

We collect 2,010,648 biomedical terminology definition pairs from three major biomedical ontology databases, including Open Biological and Biomedical Ontology Foundry (OBO) (Smith et al., 2007), BioPortal (Noy et al., 2009) and EMBL-EBI Ontology Lookup Service (OLS) (Jupp et al., 2015),

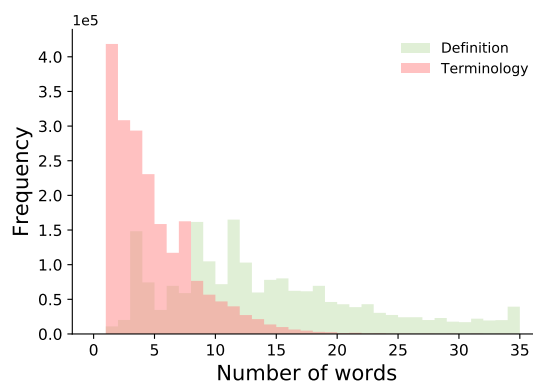


Figure 2: Bar plot showing the comparison between the number of words in the definition and in the terminology in Graphine.

spanning diverse biomedical subdisciplines such as cellular biology, molecular biology and drug development. For the definition that span multiple sentences, we only consider the first sentence.

Even though these large-scale terminology definition pairs have already presented a novel resource for definition generation, one unique feature of our dataset is the graphs among terminologies. In particular, we construct a DAG for each biomedical subdiscipline using ‘is a’ relationship from the original data. As a result, each terminology belongs to one DAG, where the node is associated with a terminology and its definition and the edge links from a general terminology to a specific one. We reduce the number of DAGs from 499 to 227 by merging DAGs that appear in more than one database.

We notice substantial amount of missing definitions in the original collection, confirming the importance of computationally generating definition. In 81 out of 499 DAGs, more than 50% of terminologies does not have any definition. We thus exclude terminologies that do not have a curated definition. We further observed a substantial discrepancy between the number of words in the terminology and the number of words in the definition. The average number of words in the terminology is 4.55, which is much lower than the 15.58 average number of words in the definition (**Figure 2**). This discrepancy could pose great challenges to text generation model. We seek to alleviate it using graph neighbor’s terminology and definition.

2.2 Data analysis

All definitions in our datasets are curated by domain experts, assuring the high-quality. Reassur-

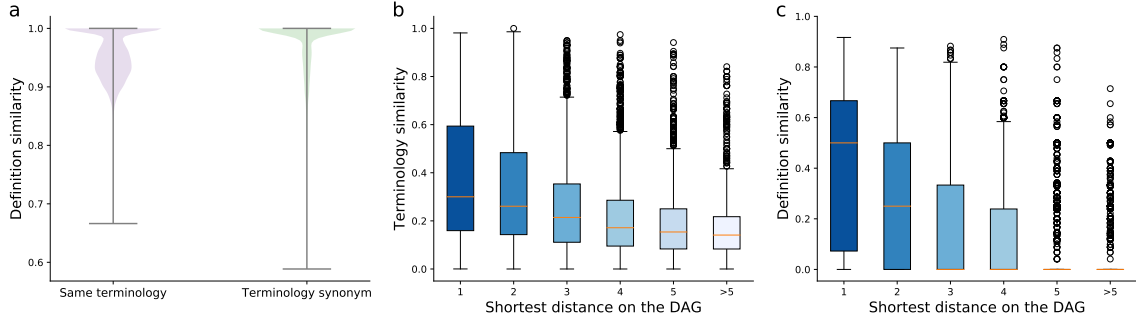


Figure 3: Analysis of Graphine. a, Violin plot showing the definition similarity between the same terminology and the terminology synonym curated by different experts. b,c, Box plots showing the terminology similarity (b) and the definition similarity (c) between nodes of different shortest distances on the DAG.

ingly, we investigate the consistency between expert curation by comparing the definitions of the same terminology from different DAGs (e.g., *material maintenance* appears in both *obi* and *chmo*). Different DAGs are curated by different domain experts in our dataset. We observed a remarkable cosine similarity of 0.96 between definitions of the same terminology (Figure 3a). We next examine the definitions of 67,257 terminology synonym pairs that presents in different DAGs. Synonyms are also curated by domain experts in the original databases. We again observed prominent cosine similarity 0.97, assuring the consistency between expert curation.

To examine the quality of the graph structure, we study the consistency between graph-based terminology similarity and text-based terminology similarity. Graph-based terminology similarity is calculated using the shortest distance on the graph. Text-based similarity is calculated using BLEU score (Papineni et al., 2002) between two terminologies. We observed strong agreement between these two similarity scores (Figure 3b). This agreement is even more substantial between graph-based terminology similarity and text-based definition similarity (Figure 3c). Collectively, these results indicate that nearby nodes exhibit similar terminologies and definitions, suggesting the opportunity to improve definition generation using the graph structure.

3 Graph-aware Definition Generation: Task and Model

3.1 Problem Definition

Our goal is to generate the definition text according to the terminology text. Meanwhile, terminologies form a DAG, which could be used to assist definition generation. More precisely, the input is a

directed acyclic graph $G = (V, E, T, D)$, where $V = \{v_i\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. Each node v_i is associated with a terminology $t_i \in T$ and a definition $d_i \in D$. t_i and d_i are both token sequences defined as $t_i \triangleq \langle t_i^1, t_i^2, \dots, t_i^{n_{t_i}} \rangle$ and $d_i \triangleq \langle d_i^1, d_i^2, \dots, d_i^{n_{d_i}} \rangle$, where $t_i^j \in C$, $d_i^j \in C$ and C is the vocabulary. In practice, the terminology is often a phrase and the definition is a sentence. Therefore, n_{d_i} is much larger than n_{t_i} .

We consider a transductive learning setting where V composes of V_{train} and V_{test} . V_{train} is the set of nodes that have both terminologies and definitions. V_{test} is the set of nodes that only have terminologies. The goal of graph-aware definition generation is to generate d_i for $v_i \in V_{test}$ according to both the terminology t_i and the graph G . Although each graph G in Graphine is a DAG, our method can be applied to any kind of graphs.

The proposed definition generation task is distinct from conditional text generation and machine translation due to the presence of this graph G . G makes it possible to transfer knowledge between terminologies based on our previous observation that nearby nodes on the graph have similar definitions. We thus aim at propagating terminology and definition using the graph structure to enhance definition generation.

3.2 Model

We propose a graph-aware definition generation approach Graphex that generates definition based on the global semantic embedding and the local semantic embedding using a two-stage approach (Fig. 4). At the first stage, global semantic embeddings are calculated through propagating terminology and definition on the graph. At the second stage, the lo-

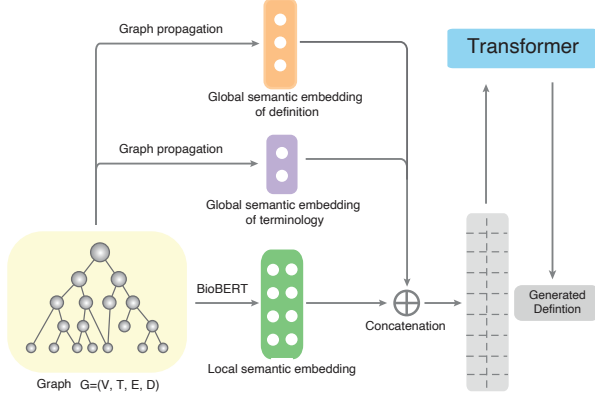


Figure 4: Flowchart of Graphex. Graphex considers the graph structure during definition generation by concatenating the global semantic embeddings and the local semantic embedding.

cal semantic embedding is obtained by embedding the specific terminology. Finally, Graphex generates the definition d_i by using the concatenation of global and local semantic embeddings as the input to a Transformer (Vaswani et al., 2017).

3.2.1 Encoding global semantic via graph propagation

At the first stage, we obtain two global semantic embeddings g_i^t and g_i^d of each node v_i through propagating terminology and definition on the graph, respectively. In particular, we follow a previous work (Kotitsas et al., 2019) to calculate g_i^t and g_i^d using a bidirectional GRU-based neural network, which aggregates the embeddings of individual words in t_i as the node features of the node v_i and then smooths node features based on random walk.

To encode the network structure, we sample m random walk paths of fixed length k starting from each node (Grover and Leskovec, 2016). The r -th random walk starting from the node v_i is denoted as $P_{v_i,r} = \langle p_{1,r}, p_{2,r}, \dots, p_{k,r} \rangle$ ($r = 1, \dots, m$), where $p_{1,r} = v_i$. We then learn two embeddings w_i and u_i for each node v_i based on the arriving probability calculated from these sampled random walk paths. In particular, the predicted probability of arriving the node v_j through the walk $P_{v_i,r}$ is defined as:

$$p(v_j|v_i) = \frac{\exp(\mathbf{u}_j^T \mathbf{w}_i)}{\sum_{v_k \in V} \exp(\mathbf{u}_k^T \mathbf{w}_i)}. \quad (1)$$

Here, w_i is the feature embedding and u_i is the context embedding for node v_i .

Instead of training w_i and u_i solely based on the network structure, we use text feature from t_i to regularize them. We define $q(c)$ and $h(c)$ to be the

two separate trainable word embeddings for each token c in the vocabulary C . Then $q(t_i^k)$ and $h(t_i^k)$ are the trainable word embeddings of the k -th token in the terminology t_i . We use a shared bidirectional GRU network to encode t_i into u_i and w_i as:

$$\mathbf{u}_i^f = GRU_f(\mathbf{q}(t_i^1), \dots, \mathbf{q}(t_i^n)) \quad (2)$$

$$\mathbf{u}_i^b = GRU_b(\mathbf{q}(t_i^1), \dots, \mathbf{q}(t_i^n)) \quad (3)$$

$$\mathbf{u}_i = Max_pooling(\mathbf{u}_i^b + \mathbf{u}_i^f) \quad (4)$$

$$\mathbf{w}_i^f = GRU_f(\mathbf{h}(t_i^1), \dots, \mathbf{h}(t_i^n)) \quad (5)$$

$$\mathbf{w}_i^b = GRU_b(\mathbf{h}(t_i^1), \dots, \mathbf{h}(t_i^n)) \quad (6)$$

$$\mathbf{w}_i = Max_pooling(\mathbf{w}_i^f + \mathbf{w}_i^b). \quad (7)$$

The loss function at the first stage is defined as:

$$L_1 = - \sum_{v_i \in V} \sum_{r=1}^m \sum_{j=2}^k \log p(v_{j,r} | v_{1,r} = v_i) \quad (8)$$

After minimizing this loss function, g_i^t is obtained by concatenating w_i and u_i , which represents the global semantic of node v_i using the terminology. Likewise, we can obtain g_i^d by first encoding d_i into the feature embedding w_i^d and the context embedding u_i^d , and then concatenating them. For node that does not have the definition (i.e., $v_i \in V_{test}$), we generate a d_i' as replacement by using t_i as input to a Transformer trained on other terminology definition pairs.

3.2.2 Fusing local and global semantic for definition generation

At the second stage, we generate the definition d_i for node v_i conditioned on both the local semantic l_i and the global semantic g_i^t and g_i^d . The local semantic l_i is obtained by embedding t_i using BioBERT (Lee et al., 2020). We also examined other BERT-based models in the experiments. Let $P(d_i | l_i, g_i^t, g_i^d; \theta)$ be the transformer model parameterized by θ . The loss function at the second stage is defined as

$$L_2 = - \sum_{i \in |V|} \log P(d_i | l_i, g_i^t, g_i^d; \theta). \quad (9)$$

4 Experimental Results

4.1 Experimental setup

We conduct experiments using DAGs included in the OBO database. To study the effect of graph structures, we only consider graphs that show a high correlation between the graph-based similarity and the text-based definition similarity as measured

in Section 2.2. Only definitions of the training data are used to calculate the correlation. We split the terminology definition pairs into 70% training, 10% validation and 20% test. The data split and model training are done within each DAG separately.

We compare our method with three conventional conditional text generation models: **Seq2Seq** (Bahdanau et al., 2014), **CVAE** (Yan et al., 2016) and **Transformer** (Vaswani et al., 2017). All of them take the terminology as the input and the definition as the output. Since none of them considers the graph structure, our comparison could reveal the importance of considering graph structures. We further implement two variants of our model to investigate the impact of propagating definition on the graph and propagating terminology on the graph. In particular, **Our Model w/o TG** is the Graphex framework that does not incorporate the terminology-derived global semantic embedding g_i^t in eq. 9. **Our Model w/o DG** is the Graphex framework that does not incorporate the definition-derived global semantic embedding g_i^d in eq. 9.

We used the same pretrained language model for all the competing methods. We chose BioBERT as it achieved the best performance among different pretrained language models. LSTM is used as the encoder and the decoder of **Seq2seq** and **CVAE** and the dimensions of the word embedding and the hidden state are set to 768. The dimensions of the word embedding and the hidden state of **Transformer** are also set to 768. In our model, we used the default hyperparameters in (Kotitsas et al., 2019) in the first stage and use the same structure as **Transformer** baseline in the second stage. The dimensions of g_i^d and g_i^t are 768. All the models were trained using the same data splits.

We used Graphex as a benchmark to compare pretrained language models on Graphine. We use **BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), **SciBERT** (Beltagy et al., 2019), **PubMedBERT** (Gu et al., 2020) and **BioBERT** (Lee et al., 2020) to provide the pretrained word embeddings for Graphine respectively. **SciBERT** fine-tunes BERT on scientific data. **PubMedBERT** and **BioBERT** are domain-specific BERTs on biomedical domain. The word embedding dimensions are all set to 768.

We compare different graph embedding methods, **GCN** (Kipf and Welling, 2016), **HGCN** (Chami et al., 2019a) and **GAT** (Veličković et al., 2018) on our dataset. The AUC and AP of link prediction are

used to evaluate the quality of graph embedding. We compare the three graph neural network methods with Euclidean embeddings and Poincare embeddings methods, **Euclidean** and **PoincareBall** (Nickel and Kiela, 2017), and feature-based methods, **HNN** (Chami et al., 2019b) and **MLP**. We follow the default hyperparameter settings in (Chami et al., 2019a)

We perform both automatic evaluation and human evaluation. For automatic evaluation, we used six standard metrics including **BLEU1-4** (Papineni et al., 2002), **METEOR** (Banerjee and Lavie, 2005) and **NIST** (Doddington, 2002). **BLEU1-4** measures the n-gram overlap between the generated sentence and the target sentence. **METEOR** improves BLEU by considering synonyms when comparing unigrams and using F1 instead of precision. **NIST** reweights words by frequency when matching n-grams to adjust the contribution of common words like "is". For human evaluation, we recruited 3 annotators to score the generated sentences of each method for 50 terminologies. Annotators are requested to grade each generated definition as 0 (bad), 1 (fair) and 2 (good).

4.2 Graphex improves definition generation by considering the graph structure

We first evaluated the performance of definition generation by Graphex. We compared Graphex with baselines that do not consider the graph structure (Table 1). We found that Graphex, which uses both the definition graph and the terminology graph, obtained the best performance on all six metrics. The improvement is most prominent against baselines that do not use the graph structure. For example, Graphex obtained 34.35 BLEU1 score, which is 7.65% and 59.62% higher than Transformer and Seq2seq. Moreover, we observed decreased performance when only the terminology graph (Our Method w/o DG) or the definition graph (Our Method w/o TG) is considered. Despite less superior performance, these two variants are still consistently better than baselines that do not use graphs, confirming the importance of modeling graph structures in definition generation.

We showed two examples of how the graph can help Graphine generate better definition (Table 2). In both examples, the true definition of the nearby node is included in the training set, and can thus be used to capture the global semantic. We found that Graphex selectively copied tokens in the true

Model	TG	DG	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	NIST	Human
Seq2Seq			21.52	14.47	10.82	8.56	9.42	0.69	0.87
CVAE			20.05	13.48	10.02	8.23	8.97	0.67	0.83
Transformer			31.91	25.09	21.26	18.70	15.81	1.06	1.01
Our Model w/o TG		✓	33.81	26.23	22.16	19.26	16.55	1.10	1.09
Our Model w/o DG	✓		32.47	25.52	21.73	19.30	16.17	1.11	1.06
Our Model	✓	✓	34.35	26.97	22.99	20.21	16.57	1.15	1.12

Table 1: Comparison on the performance of definition generation using automatic and human evaluation. TG (DG) refers to propagation on terminologies (definitions).

Terminology:	estuarine tidal riverine open water pycnocline
True definition:	an estuarine open water pycnocline which is composed primarily of fresh tidal water
Parent definition:	a pycnocline which is part of an estuarine water body, spanning from a fiat boundary where the estuary bed below the water column reaches a depth of 4 meters until the end of the estuary most distal from the coast
Graphex:	an estuarine water which extends from an estuarine pycnocline or mid - depth to the estuary bed and from a fiat boundary where the estuary bed below the water column
Transformer:	an area of a planet’s surface which is primarily covered by UNK herbaceous vegetation and where the underlying soil or
Terminology:	increased eye tumor incidence
True definition:	greater than the expected number of tumors originating in the eye in a given population in a given time period
Child definition:	greater than the expected number of neoplasms in the retina, usually in the form of a distinct mass, in a specific population in a given time period
Graphex:	greater than the expected number of neoplasms in the gastric tissue usually in the form of a distinct mass , in a specific population in a given time period
Transformer:	greater than the expected number of UNK in the lung , usually in the form of a distinct mass

Table 2: Comparison between definitions generated by Graphex and the best baseline Transformer. True definitions of the nearby node are also listed to illustrate the effect of considering graph structures.

definition of the parent node, leading to a more accurate generation. For example, in the first case, Graphex successfully generated *estuarine water* and *estuary bed below the water column*. In the second case, Graphex propagated *in a given population in a given time period* from the child node, resulting in the correct generation of *in a given population in a given time*. In contrast, the Transformer baseline is not able to generate such detailed information in both examples due to the ignorance of graph structures. Since the pretrained language model and the graph representation method are two important model selections in Graphex, we next leverage Graphex to compare different pretrained language models and graph representation methods, shedding light on future directions in definition generation.

4.3 Comparing Pretrained Language Models

Domain-specific pretrained language models have achieved impressive performance on tasks such as named entity recognition, information extraction and relation extraction in biomedicine (Beltagy et al., 2019; Lee et al., 2020). One barrier to more thoroughly comparing these pretrained language

Pretrain	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	NIST
BERT	30.83	23.72	19.95	17.50	14.70	1.02
RoBERTa	25.12	18.36	14.95	12.73	12.17	0.80
SciBERT	33.95	26.55	22.53	19.96	16.21	1.15
PubMedBERT	31.35	24.31	20.57	18.01	15.12	1.05
BioBERT	34.35	26.97	22.99	20.21	16.57	1.15

Table 3: Comparison on the performance of definition generation using different pretrained language models. SciBERT, PubMedBERT, BioBERT are domain-specific pretrained language models.

models is the lack of domain-specific benchmarks. Graphine could be used as a novel domain-specific benchmark in biomedicine. As a proof-of-concept, we compared five pretrained language models, including three biomedical domain-specific models, by using it to generate the local semantic l_i in eq. 8 (Table 3). We found that domain-specific pretrained language models have consistently better performance than general pretrained language models, which agrees with previous findings on the value of domain-specific language models in biomedicine (Gu et al., 2020; Lee et al., 2020; Beltagy et al., 2019). Within the three domain-specific pretrained language models, BioBERT and SciBERT obtained the most prominent performance. This might be due to the corpus these two models

Model	AUC	AP
Euclidean	0.8979	0.9307
PoincareBall	0.9069	0.9346
HNN	0.9023	0.9211
MLP	0.8892	0.9258
GCN	0.9493	0.9659
HGCN	0.8996	0.9365
GAT	0.8867	0.9179

Table 4: Comparison on the performance of link prediction using different graph representation learning methods.

were trained on, suggesting the possibility to use Graphine to further compare different biomedical corpus (Wang et al., 2020a; Lo et al., 2020).

4.4 Comparing graph representation methods

We next sought to compare graph representation methods using a link prediction task based on our dataset. Graphs in our dataset present a hierarchical structure, which poses a unique challenge for graph representation methods. The results are summarized in **Table 4**. We found that methods that consider the graph structure have overall superior performance, conforming the importance of the graph structure in definition generation. Among all approaches, GCN obtained the best performance. We didn’t observe improved performance by embedding graphs into the hyperbolic space, which is contradictory to prior work showing that hyperbolic embedding can better model hierarchical structures (Nickel and Kiela, 2017; Chami et al., 2019a). We attribute this to the more complicated node features in contrast to previous work. In our dataset, node features are text that has arbitrary length and large vocabulary, introducing new challenges to hyperbolic embedding-based methods.

4.5 Sentence granularity prediction

Finally, we exploited Graphine for a novel task of sentence granularity prediction. Measuring sentence semantic similarity is crucial for many NLP tasks. Existing sentence similarity benchmarks only provide binary labels indicating similar or dissimilar (Li et al., 2006; Mueller and Thyagarajan, 2016). In contrast, our dataset is able to characterize the specific granularity of sentences beyond similarity. We define the ground truth granularity of a definition sentence as its depth in the DAG, where a smaller (larger) depth indicates a more coarse-grained (fine-grained) sentence. Based on this granularity benchmark, we define two specific

tasks: relative granularity prediction and absolute granularity prediction. Relative granularity prediction aims at predicting which sentence is more fine-grained between two given sentences. Absolute granularity prediction aims at predicting the specific granularity of a given sentence. The incomparable granularity levels from different graphs could introduce systematic bias to comparing sentences from different graphs. To tackle this problem, we first performed a graph alignment among all DAGs using terminologies that appeared in multiple DAGs as anchors. After the alignment, all sentences are associated with a granularity level between 1 and 17, where 1 indicates the most coarse-grained sentence.

To predict the relative granularity, we used the concatenation of the BERT embeddings of two sentences as features to train an multi-layer perceptron (MLP). When comparing sentences within the same DAG, 76% of graphs obtained an accuracy larger than 0.80 (**Figure 5a**). We next examined the accuracy of classifying a pair of sentences from two different DAGs and also observed a good accuracy of 0.81. To predict the absolute granularity, we used the BERT embedding of each sentence as features to train an MLP-based multi-class classifier. We again observed desirable accuracy of 0.71 and 0.81 and Spearman correlation 0.60 and 0.69 within each graph and across all graphs (**Figure 5b,c**). In addition to predicting sentence granularity, we envision this new benchmark of sentence granularity could provide deeper insight into evaluating existing sentence similarity models through transforming it from a binary classification task to a ranking task.

5 Future work motivated by an opposite generation

Despite the overall improved performance of Graphex, we found that some definitions generated by Graphex present an opposite meaning to the truth definition. We showed one of such example in **Figure 6**. Although the definition generated by Graphex for *hyperlasia* matches the true definition well, the generated definition has the opposite semantic meaning (e.g., *reduction*, *reduced*) to the true definition (e.g., *increase*, *increased*). Notably, such failed generations cannot be captured by existing n -gram based metrics, leading to artificial improvement. After a closer examination, we found that this opposite generation is caused by using

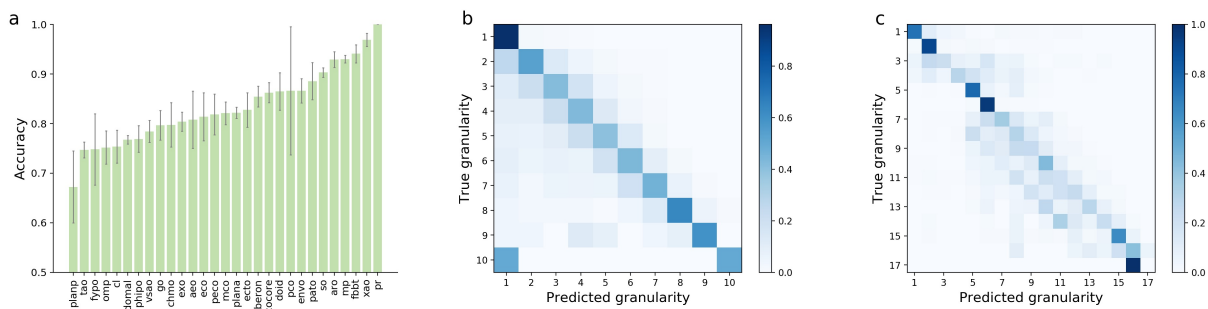


Figure 5: Sentence granularity prediction. a, Bar plot showing the accuracy of relative granularity prediction within each DAG. b,c, Heatmaps showing the accuracy of absolute granularity prediction within each DAG (b) and across all DAGs (c).

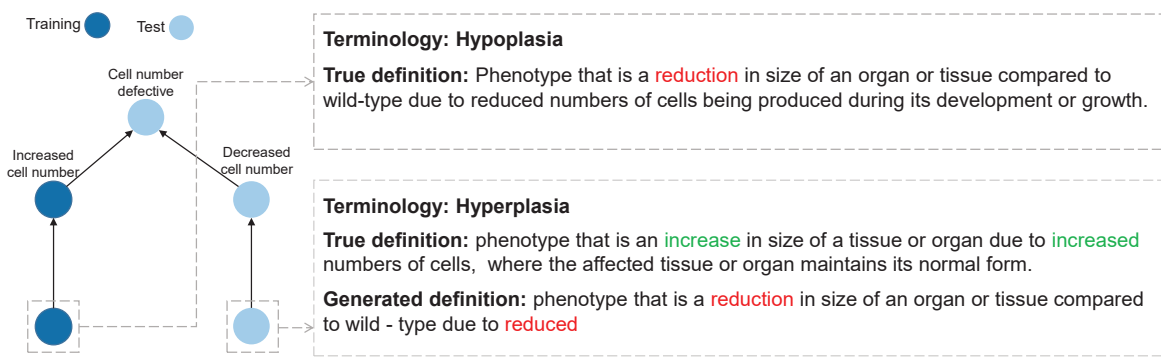


Figure 6: A failed generation that cannot be captured by existing evaluation metrics. Graphex generated a sentence that has the opposite meaning to the true definition.

the definition from a cousin node *hypolasia* in the graph. Moreover, existing BERT-based models are not able to effectively associate subword *hypo* (*hyper*) in the terminology with *reduce* (*increase*) in the definition. We plan to explore the possibility of developing faithful generation models (Wang et al., 2020d) to address this problem and leave it as an important future work.

6 Relate Work

Existing works related to terminology definition mainly focus on definition extraction (Westerhout, 2009; Anke and Schockaert, 2018; Veyseh et al., 2020; Li et al., 2016) and technology entity recognition (Fahmi and Bouma, 2006; Gao et al., 2018). Definitions are extracted from different sources, such as Wikipedia (Espinosa-Anke and Saggion, 2014; Li et al., 2016) and scholarly articles (Jin et al., 2013; Spala et al., 2019). In contrast to previous work, We study the novel problem of terminology definition generation. Notably, the proposed dataset Graphine can also be used as a new benchmark to evaluate existing approaches on extracting

definitions from the free text. Many scientific literature datasets have been curated for a variety of tasks, such as hypothesis generation (Spangler et al., 2014), scientific claim verification (Wadden et al., 2020), paraphrase identification (Vinyals et al., 2016; Dong et al., 2021; Xu et al., 2016) and citation recommendation (Saier and Färber, 2019). Paraphrase identification datasets, such as MSCOCO, Quora, MSR, ParaSCI, are most related to our work (Vinyals et al., 2016; Dong et al., 2021; Xu et al., 2016). Distinct from these datasets, we focused on a different task (i.e., definition generation) and a different domains (i.e., biomedical domain).

Graph2text and data2text, which aim at generating text from structured data, have attracted increasing attention (Marcheggiani and Perez-Beltrachini, 2018; Cai and Lam, 2020; Yao et al., 2020; Guo et al., 2020; Wang et al., 2019). Among them, AMR-to-text Generation and knowledge graph to text generation also consider graph structures. The Abstract Meaning Representation (AMR) represents the semantic information of each sentence

using a rooted directed graph, where each edge is a semantic relations and each node is a concept (Song et al., 2018; Zhu et al., 2019; Mager et al., 2020; Wang et al., 2020c). Knowledge graph to text generation has advanced tasks such as entity description generation and medical image report by generating text from a subgraph in the knowledge graph (Cheng et al., 2020; Li et al., 2019). Despite all considering graph structures, our method generates one sentence for each node on a large directed acyclic graph, whereas AMR-to-text and knowledge graph to text generate sentences for a subgraph or the entire graph.

7 Conclusion

We have introduced a novel dataset Graphine for studying definition generation. Graphine includes 2,010,648 terminology definition pairs from three major biomedical databases. Terminologies in Graphine form 227 directed acyclic graphs, which make Graphine a unique resource for exploring graph-aware text generation. We have proposed a graph-aware definition generation method Graphex, which takes the graph structure into consideration. Graphex has obtained substantial improvement against methods that do not consider graph structures. Moreover, we have illustrated how Graphine can be used to evaluate other tasks, including comparing pretrained language models, comparing graph representation learning methods and predicting sentence granularity. Finally, we have analyzed the definition generated by our method and proposed future directions to improve. Collectively, we envision our dataset to be a unique resource for definition generation and could be broadly utilized by other natural language processing applications.

Acknowledgement

This paper is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101900/2018AAA0101902 as well as the National Natural Science Foundation of China (NSFC Grant No. 62106008 and No. 61772039).

References

David Altshuler, Mark Daly, and Leonid Kruglyak. 2000. Guilt by association. *Nature genetics*, 26(2):135–137.

- Luis Espinosa Anke and Steven Schockaert. 2018. Syntactically aware neural architectures for definition extraction. In *NAACL*, pages 378–385.
- Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Abdul Mannan Baig. 2020. Chronic covid syndrome: Need for an appropriate medical terminology for long-covid and covid long-haulers. *Journal of medical virology*.
- Darrin Baines, RJ Elliott, et al. 2020. Defining misinformation, disinformation and malinformation: An urgent need for clarity during the covid-19 infodemic. *Discussion Papers*, 20.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. pages 3615–3620.
- Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. pages 10–21.
- Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *AAAI*, volume 34, pages 7464–7471.
- Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. 2019a. Hyperbolic graph convolutional neural networks. *NeurIPS*, page 4869.
- Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019b. Hyperbolic neural networks. *NeurIPS*, pages 4868–4879.
- Liyang Cheng, Dekun Wu, Lidong Bing, Yan Zhang, Zhanming Jie, Wei Lu, and Luo Si. 2020. Ent-desc: Entity description generation by exploring knowledge graph. *arXiv preprint arXiv:2004.14813*.
- James J Cimino, Paul D Clayton, George Hripcsak, and Stephen B Johnson. 1994. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *Journal of the American Medical Informatics Association*, 1(1):35–50.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT*, pages 138–145.

- Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. Parasci: A large scientific paraphrase dataset for longer paraphrase generation. pages 424–434.
- Luis Espinosa-Anke and Horacio Saggion. 2014. Applying dependency relations to definition extraction. In *NLDB*, pages 63–74.
- Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*.
- Hui Gao, Ting Wang, Wei Luo, and Lin Gui. 2018. Adversarial multitask learning for technology entity recognition. In *PIC*, pages 134–139.
- Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*.
- Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. pages 77–88.
- Dishan Gupta, Jaime Carbonell, Anatole Gershan, Steve Klein, and David Miller. 2015. Unsupervised phrasal near-synonym generation from text corpora. In *AAAI*, volume 29.
- Yiping Jin, Min-Yen Kan, Jun Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the acl anthology. In *EMNLP*, pages 780–790.
- Simon Jupp, Tony Burdett, Catherine Leroy, and Helen E Parkinson. 2015. A new ontology lookup service at embl-ebi. In *SWAT4LS*, pages 118–119.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Sotiris Kotitsas, Dimitris Pappas, Ion Androutsopoulos, Ryan McDonald, and Marianna Apidianaki. 2019. Embedding biomedical ontologies by jointly encoding network structure and textual node descriptors. pages 298–308.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, pages 1234–1240.
- Christy Y Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. 2019. Knowledge-driven encode, retrieve, paraphrase for medical image report generation. In *AAAI*, pages 6666–6673.
- Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael R Lyu, and Irwin King. 2020. Unsupervised text generation by learning from search. *arXiv preprint arXiv:2007.08557*.
- SiLiang Li, Bin Xu, and Tong Lee Chung. 2016. Definition extraction with lstm recurrent neural networks. In *CCL*, pages 177–189. Springer.
- Yuhua Li, David McLean, Zuhair A Bandar, James D O’shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, pages 1138–1150.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2orc: The semantic scholar open research corpus. pages 4969–4983.
- 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. pages 1846–1852.
- Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. pages 90–98.
- Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. pages 1–9.
- Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, volume 30.
- Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *NeurIPS*, 30:6338–6347.
- Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, Christopher G Chute, et al. 2009. Biportal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, pages W170–W173.
- Tim R Oke. 2006. Towards better scientific communication in urban climate. *Theoretical and Applied Climatology*, 84(1):179–190.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.
- Tarek Saier and Michael Färber. 2019. Bibliometric-enhanced arxiv: A data set for paper-based and citation-based tasks. In *BIR@ ECIR*, pages 14–26.
- Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. 2007. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, pages 1251–1255.
- Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric P Xing. 2020a. Generalized zero-shot text classification for icd coding. In *IJCAI*.
- Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. pages 1616–1626.
- Yiping Song, Zequn Liu, Wei Bi, Rui Yan, and Ming Zhang. 2020b. Learning to customize model structures for few-shot dialogue generation tasks. In *ACL*, pages 5832–5841.
- Sasha Spala, Nicholas A Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. Deft: A corpus for definition extraction in free-and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131.
- Scott Spangler, Angela D Wilkins, Benjamin J Bachman, Meena Nagarajan, Tajhal Dayaram, Peter Haas, Sam Regenbogen, Curtis R Pickering, Austin Comer, Jeffrey N Myers, et al. 2014. Automated hypothesis generation based on mining scientific literature. In *SIGKDD*, pages 1877–1886.
- 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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeuralPS*, pages 5998–6008.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks.
- Amir Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. In *AAAI*, volume 34, pages 9098–9105.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. pages 7534–7550.
- Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, et al. 2020a. Cord-19: The covid-19 open research dataset.
- Sheng Wang, Hyunghoon Cho, ChengXiang Zhai, Bonnie Berger, and Jian Peng. 2015. Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, 31(12):i357–i364.
- Sheng Wang, Jianzhu Ma, Samson Fong, Stefano Rensi, Jiawei Han, Jian Peng, Dexter Pratt, Russ B Altman, and Trey Ideker. 2019. Deep functional synthesis: a machine learning approach to gene functional enrichment. *bioRxiv*, page 824086.
- Sheng Wang, Angela Oliveira Pisco, Aaron McGeever, Maria Brbic, Marinka Zitnik, Spyros Darmanis, Jure Leskovec, Jim Karkaniyas, and Russ B Altman. 2020b. Unifying single-cell annotations based on the cell ontology. *bioRxiv*, page 810234.
- Tianming Wang, Xiaojun Wan, and Shaowei Yao. 2020c. Better amr-to-text generation with graph structure reconstruction. In *IJCAI*, pages 3919–3925.
- Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020d. Towards faithful neural table-to-text generation with content-matching constraints. pages 1072–1086.
- Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 61–67.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296.
- Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, pages 776–791. Springer.
- Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *ACL*, pages 7145–7154.
- 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. pages 5459–5468.