Automatic Term Name Generation for Gene Ontology: Task and Dataset

Yanjian Zhang[1], Qin Chen[1]*, Yiteng Zhang[1], Yixu Gao[1], Zhongyu Wei[15], Jiajie Peng[2], Zengfeng Huang[1], Weijian Sun[3], Xuanjing Huang[4]
[1]School of Data Science, Fudan University
[2]School of Computer Science, Northwestern Polytechnical University
[3]Huawei Technologies Co., Ltd
[4]School of Computer Science, Fudan University
[5]Research Institute of Intelligent and Complex Systems, Fudan University
{yanjianzhang16,qin_chen,yitengzhang19,yxgao19,zywei,huangzf,xjhuang}@fudan.edu.cn
jiajiepeng@nwpu.edu.cn
sunweijian@huawei.com

Abstract

Terms contained in Gene Ontology (GO) have been widely used in biology and bio-medicine. Most previous research focuses on inferring new GO terms, while the term names that reflect the gene function are still named by the experts. To fill this gap, we propose a novel task, namely term name generation for GO, and build a large-scale benchmark dataset. Furthermore, we present a graph-based generative model that incorporates the relations between genes, words and terms for term name generation, which exhibits great advantages over the strong baselines.

1 Introduction and Related Work

Gene Ontology (GO) is a widely-used biological ontology, which contains a large number of terms to describe the gene function in three aspects, namely molecular function, biological process and cellular component (Consortium, 2015, 2016). The terms are organized hierarchically like a tree, and can be used to annotate the genes as demonstrated in Figure 1. GO has been extensively studied in the research community of bio-medicine and biology for its great value in many applications, such as protein function analysis (Cho et al., 2016) and disease association prediction (Menche et al., 2015).

A major concern in GO is the GO construction, including term discovery, naming and organization (Mazandu et al., 2017; Koopmans et al., 2019). In early studies, the terms are manually defined and organized by the experts in particular areas of biology, which is very labor-consuming and inefficient given the large volume of biological literature published every year (Tomczak et al., 2018). Moreover, different experts may use different expressions to describe the same biological concept, causing an inconsistency problem in term naming.

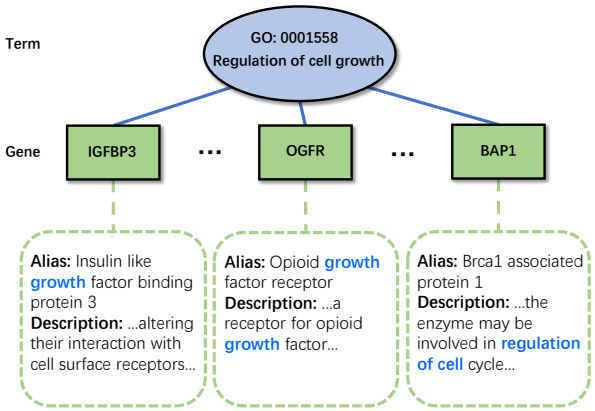

Figure 1: A term named "Regulation of cell growth" and the related genes with aliases and descriptions.

Recently, many researchers turn to develop automatic methods for GO construction. Dutkowski et al. (Dutkowski et al., 2013) proposed a Network-eXtracted Ontology (NeXO), which clustered genes hierarchically based on their connections in the molecular networks, and recovered around 40% of the terms according to the alignment between NeXO and GO. In order to further improve the performance, Kramer et al. (Kramer et al., 2014) identified the gene cliques which were treated as a term in an integrated biological network. Though these methods infer new GO terms and their relationships based on the structured networks automatically (Gligorijević et al., 2014; Li and Yip, 2016; Peng et al., 2015), the new terms are still named manually by the experts, which is prone to the problems of inefficiency and inconsistency. Furthermore, only the structure information in existing networks is utilized, while the genes' rich textual information that potentially describes the corresponding term has not well been studied.

In order to obtain term names automatically to boost GO construction, we propose a novel task that aims to generate term names based on the textual information of the related genes. An illustrative example of the task is shown in Figure 1. The

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4705–4710

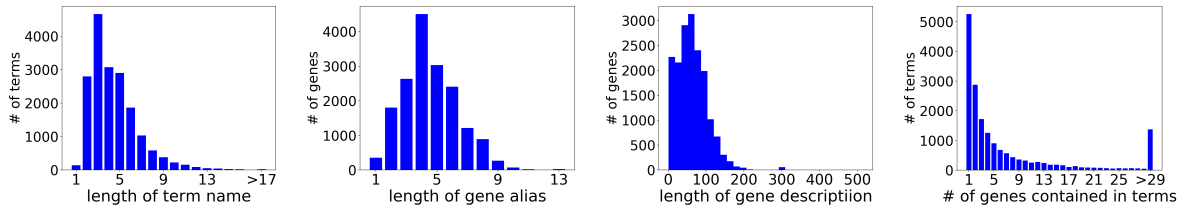

Figure 2: Distributions of the dataset.

genes IGFBP3, OGFR and BAP1 are annotated by the term with the ID as GO:0001558 and name as "Regulation of cell growth". Since there are some word overlaps between the term name and gene text (alias and description) by our observations, we aim to generate the term name based on the gene text. To facilitate the research, we first present a dataset for term name generation in GO. Then, we propose a graph-based generative model that incorporates the potential relations between genes, words and terms for term name generation. The experimental results indicate the effectiveness of our proposed model. The contributions of our work are threefold: (1) To the best of our knowledge, it is the first attempt to explore to generate term names for GO automatically. (2) We present a large-scale dataset for term name generation based on various biological resources, which will help boost the research in bio-medicine and biology. (3) We conduct extensive experiments with in-depth analyses, which verify the effectiveness of our proposed model.

2 Dataset

We build a large-scale dataset[1] for term name generation, which contains the GO terms about Homo sapiens (humankind). We collect the term ID, term name and the corresponding genes' ID from Gene Ontology Consortium[2]. In addition, the gene alias and descriptions are crawled from GeneCards[3], which contains the information from Universal Protein Resource (UniProt)[4].

Our dataset contains 18,092 samples in total. Each sample contains a term ID, term name and the related genes with alias and descriptions as demonstrated in Figure 1. The statistics and distributions about the dataset are shown in Table 1 and Figure 2. We observe that about 51.3% of the words are shared between term names and related genes, indicating the potential to utilize the textual information of genes for term name generation. It is also interesting to find that some patterns like "*regulation of*" appear in the term name frequently, which provide valuable clues for enhancing the performance of generation.

# of terms	18,092
# of genes	17,233
Avg. length of term name	4.74
Avg. length of gene alias	4.83
Avg. length of gene description	66.1
Shared words between term and gene	51.3%

Table 1: Statistics of the dataset.

3 Graph-based Generative Model

The classical generative models such as Seq2Seq (Sutskever et al., 2014), HRNNLM (Lin et al., 2015) and Transformer (Vaswani et al., 2017) only incorporate the sequential information of the source text for sentence generation, while the potential structure within the text is neglected. To alleviate this problem, we build a heterogeneous graph with the words, genes and terms as nodes, and adopt a graph-based generative model for term name generation. The overall architecture of our graph-based generative model is shown in Figure 3, which consists of two components: the GCN based encoder and the graph attention based decoder.

3.1 GCN based Encoder

The GCN-based encoder aims to encode the relations between genes, words and terms for boosting term name generation. We first construct a heterogeneous graph based on the dataset, and then apply the Graph Convolutional Network (GCN) (Vashishth et al., 2019) for representation learning.

Graph Construction. We build a heterogeneous graph where the nodes are the words, genes and terms, and the edges reflect the relations between them. The words come from the gene text. Regarding to the edges, there are two types: word-gene and gene-term. The value for the word-gene edge is the normalized count of the word in the

[1] https://www.disc.fudan.edu.cn/data/fudan_term_name_generation.zip
[2] http://geneontology.org/
[3] https://www.genecards.org/
[4] https://www.uniprot.org/

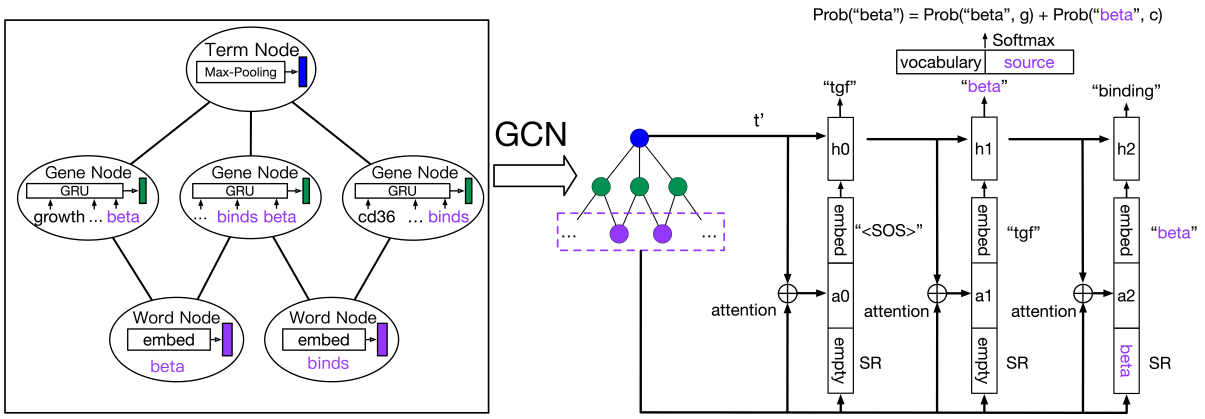

Figure 3: The overall architecture of our Graph-based Generative Model. Prob("beta", g) and Prob("beta", c) denote the probabilities based on the generation-mode and copy-mode respectively.

gene text, while the value for the gene-term edge is 1 if the gene can be annotated by the term.

Representation Learning. The initial representation for the word node is the word embeddings. For the gene node, the gene alias and description encoded by a GRU model is used as the initial representation. Regarding to the term node, the pooling over all the representations of the related gene nodes is used as the initial representation. Then, we update the node representation via a GCN model due to its effectiveness in modeling the structure information (Kipf and Welling, 2016), which is formulated as follows:

$$\mathcal{X}' = \hat{A}\,\mathrm{ReLU}\left(\hat{A}\mathcal{X}W^{(0)}\right)W^{(1)} \tag{1}$$

where $\hat{A} = A + I$, A is the adjacency matrix of the graph, and I is the identity matrix. $\mathcal{X}$ is the initial representation for the nodes, denoted as $\mathcal{X} = (t, g_1 ... g_m, w_1, ..., w_n)$, where g_i, w_i, t denote the initial representation for the ith gene, word and term respectively. $W^{(0)}$ and $W^{(1)}$ represent the weight matrix in the first and second layer of GCN.

3.2 Graph Attention based Decoder

Motivated by the effectiveness of the attention mechanism for generation (Bahdanau et al., 2014), we adopt a graph attention based decoder to generate the term name. The attentive word node representation by GCN is utilized and formulated as:

$$a_t = \sum_{j=1}^{n} \alpha_j w'_j$$
$$\alpha_j = softmax(v^T \tanh(W_a[h_{t-1}; w'_j])) \tag{2}$$

where h_{t-1} is the previous hidden state, w'_j is the word node representation by GCN, v is a parameter vector, and W_a is a parameter matrix.

Given the word overlaps between the gene text and term name, we utilize the copy mechanism in CopyNet (Gu et al., 2016) for decoding, making it possible to generate the word from either the vocabulary of the training set or the current gene text. The initial hidden state h_0 is the term node representation (i.e., t') obtained by GCN, and the hidden state is updated as:

$$h_t = f([h_{t-1}; w_{t-1}; a_t; w'_{SR}]) \tag{3}$$

where f is the RNN function, w_{t-1} is the word embedding of the previous generated word, w'_{SR} is a selective read (SR) vector in CopyNet. When the previous generated word appears in the gene text, the next word will also probably come from it, and thus w'_{SR} is the previous word node representation; otherwise it is a zero vector.

The probability of generating a target word y_t is calculated as a mixture of the probabilities by the generation-mode and copy-mode as follows:

$$p\left(y_t|h_t\right) = \frac{1}{Z}e^{\psi_g(y_t)} + \frac{1}{Z}\sum e^{\psi_c(x_j)} \tag{4}$$

where $\psi_g\left(y_t\right)$ and $\psi_c\left(x_j\right)$ are score functions for the generate-mode and copy-mode respectively, which can be defined as demonstrated in (Gu et al., 2016). $Z = \sum_{v\in\mathcal{V}} e^{\psi_g(v)} + \sum_{x\in S} e^{\psi_c(x)}$, where $\mathcal{V}$ denotes the word vocabulary in the training set, and $\mathcal{S}$ denotes the source word set in the gene text. It is notable that there are a lot of fixed patterns in the term names as mentioned in section 2. Therefore, we extract top ranked bigrams and trigrams, and treat them as new words for ease of generation.

4 Experiment

4.1 Experimental Setup

Implementation Details. The dataset is divided into the training, validation and test sets with a proportion of 8:1:1. We adopt the widely used evaluation metrics like BLEU1-3 (Papineni et al.,

Model	Rouge-1	Rouge-2	Rouge-L	BLEU-1	BLEU-2	BLEU-3
TF-IDF	9.6	*	*	9.6	*	*
LexRank	9.7	*	*	9.7	*	*
Seq2Seq	18.8	10.0	16.0	11.7	7.4	2.5
HRNNLM	19.0	10.1	16.3	11.7	7.4	2.8
Transformer	17.7	8.7	16.7	15.0	9.1	3.9
full model	21.6	**10.3**	**22.1**	**17.8**	**10.6**	**4.0**
Ablation study						
No copy	**22.5**	10.3	20.6	17.5	10.2	3.8
No pattern	21.3	9.7	22.0	16.5	9.2	3.3
No copy and pattern	21.0	10.1	18.6	15.6	9.2	3.1

Table 2: Overall performance of different models. The best result is marked in bold. Only the Rouge-1 and BLEU-1 scores for the extractive models are shown since they usually extract the unigrams independently.

2002) and Rouge$_{1,2,L}$ (Lin, 2004) for the generation task. The word embeddings are initialized from $\mathcal{N}(0, 1)$ with a dimension of 300 and updated during training. The dimension of the hidden units for GRU (Chung et al., 2014) and GCN is 300. We initialize the parameters according to a uniform distribution with the Xavier scheme (Kumar, 2017), and the dropout rate is set to 0.5. The Adam (Kingma and Ba, 2014) method with a learning rate of 1e-3 is used for training.

Baseline Methods. To evaluate the effectiveness of our proposed model, we apply the advanced baselines in two categories for comparison: (1) TF-IDF; (2) LexRank (Erkan and Radev, 2004); (3) Seq2Seq (Sutskever et al., 2014); (4) HRNNLM (Lin et al., 2015); (5) Transformer (Vaswani et al., 2017). The former two are extractive models which extract words from the gene text as the term name, and the latter three are generative models which generate words from the vocabulary space as the term name.

4.2 Experimental Results

The experimental results are shown in Table 2. It is observed that the generative models perform better than the extractive models by incorporating the language probability into generation, which makes the generated term name more coherent. Whereas, the extractive models usually extract keywords independently, which are hard to form a complete and brief term name. It is also notable that our graph-based generative model achieves the best performance in all cases by incorporating the relations between the genes, words and terms into generation. While other generative models bring unnecessary sequential information of multiple genes, which may have a side effect on term name generation.

From the ablation study, we find that when we treat the frequent patterns as new words during generation and then restore them, the performance can be further boosted. In addition, the copy mechanism can help improve the generation performance especially in the metric of BLEU scores, which proves the effectiveness of using the shared words between genes and terms for term name generation.

4.3 Visualization of Attention

To have an insight of why our proposed graph-based generative model is more effective, we randomly sample a generated term name that is the same as the ground truth, and draw an attention heatmap for the words in the term name and the corresponding gene aliases in Figure 4. The attention result for the gene descriptions is not presented here due to the limited space. We observe that the word *Tweety* that represents a gene group[5] in gene aliases is highly related to the words as *Transporter* and *Activity* in the term name, which indicates the potential of modeling the relations between words, genes and terms for enhancing the performance of term name generation.

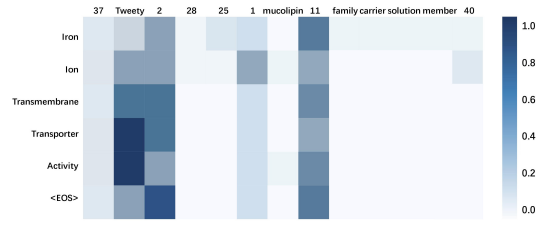

Figure 4: Attentive weight visualization. The vertical and horizontal axises denote the words in the term name and gene aliases respectively.

5 Conclusions and Future Work

In this paper, we propose a novel task of automatic term name generation based on the gene text for GO. We construct a large-scale dataset and provide the insights of this task. Experimental results show that our proposed graph-based generative model is superior to other strong baselines by modeling

[5] https://flybase.org/reports/FBgg0000560.html

the relations between genes, words and terms. In the future, we will explore how to utilize more knowledge to guide term name generation.

Acknowledgement

This work is partically supported by National Natural Science Foundation of China (No. 71991471, 61702421, 61906045), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, No.18DZ1201000, 17JC1420200), CURE (Hui-Chun Chin and Tsung-Dao Lee Chinese Undergraduate Research Endowment) (19931), China Postdoctoral Science Foundation(No.2019M661361), and National University Student Innovation Program (202010246045).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Hyunghoon Cho, Bonnie Berger, and Jian Peng. 2016. Compact integration of multi-network topology for functional analysis of genes. *Cell systems*, 3(6):540–548.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Gene Ontology Consortium. 2015. Gene ontology consortium: going forward. *Nucleic acids research*, 43(D1):D1049–D1056.

Gene Ontology Consortium. 2016. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic acids research*, 45(D1):D331–D338.

Janusz Dutkowski, Michael Kramer, Michal A Surma, Rama Balakrishnan, J Michael Cherry, Nevan J Krogan, and Trey Ideker. 2013. A gene ontology inferred from molecular networks. *Nature biotechnology*, 31(1):38.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Vladimir Gligorijević, Vuk Janjić, and Nataša Pržulj. 2014. Integration of molecular network data reconstructs Gene Ontology. *Bioinformatics*, 30(17):i594–i600.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Frank Koopmans, Pim van Nierop, Maria Andres-Alonso, Andrea Byrnes, Tony Cijsouw, Marcelo P Coba, L Niels Cornelisse, Ryan J Farrell, Hana L Goldschmidt, Daniel P Howrigan, et al. 2019. Syngo: an evidence-based, expert-curated knowledge base for the synapse. *Neuron*, 103(2):217–234.

Michael Kramer, Janusz Dutkowski, Michael Yu, Vineet Bafna, and Trey Ideker. 2014. Inferring gene ontologies from pairwise similarity data. *Bioinformatics*, 30(12):i34–i42.

Siddharth Krishna Kumar. 2017. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*.

Le Li and Kevin Y Yip. 2016. Integrating information in biological ontologies and molecular networks to infer novel terms. *Scientific reports*, 6:39237.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907.

Gaston K Mazandu, Emile R Chimusa, and Nicola J Mulder. 2017. Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in bioinformatics*, 18(5):886–901.

Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. 2015. Uncovering disease-disease relationships through the incomplete interactome. *Science*, 347(6224):1257601.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jiajie Peng, Tao Wang, Jixuan Wang, Yadong Wang, and Jin Chen. 2015. Extending gene ontology with gene association networks. *Bioinformatics*, 32(8):1185–1194.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Aurelie Tomczak, Jonathan M Mortensen, Rainer Winnenburg, Charles Liu, Dominique T Alessi, Varsha Swamy, Francesco Vallania, Shane Lofgren, Winston Haynes, Nigam H Shah, et al. 2018. Interpretation of biological experiments changes with evolution of the gene ontology and its annotations. *Scientific reports*, 8(1):1–10.

Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2019. Dating documents using graph convolution networks. *arXiv preprint arXiv:1902.00175*.

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 *Advances in neural information processing systems*, pages 5998–6008.