

MedWriter: Knowledge-Aware Medical Text Generation

Youcheng Pan^{1†}, Qingcai Chen^{1,2}, Weihua Peng³, Xiaolong Wang¹, Baotian Hu^{1*},
Xin Liu¹, Junying Chen¹, Wenxiu Zhou¹

¹Harbin Institute of Technology, Shenzhen

²Peng Cheng Laboratory

³Baidu International Technology (Shenzhen) Co., Ltd

panyoucheng4@gmail.com

{qingcai.chen, xlwangsz, hubaotian}@hit.edu.cn

pengweihua@baidu.com

Abstract

To exploit the domain knowledge to guarantee the correctness of generated text has been a hot topic in recent years, especially for high professional domains such as medical. However, most of recent works only consider the information of unstructured text rather than structured information of the knowledge graph. In this paper, we focus on the medical topic-to-text generation task and adapt a knowledge-aware text generation model to the medical domain, named MedWriter, which not only introduces the specific knowledge from the external MKG but also is capable of learning graph-level representation. We conduct experiments on a medical literature dataset collected from medical journals, each of which has a set of topic words, an abstract of medical literature and a corresponding knowledge graph from CMeKG. Experimental results demonstrate incorporating knowledge graph into generation model can improve the quality of the generated text and has robust superiority over the competitor methods.

1 Introduction

Medical text generation has been a hot topic recently, such as electronic medical record (EMR) generation (Guan et al., 2018), medical question generation (Zhang et al., 2018), clinical notes generation (Melamud and Shivade, 2019), etc. However, compared to the research in the general domain, there is still a lot of space for exploration, especially with the assistance of specific knowledge graph.

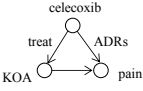
topic	塞来昔布, 骨创伤治疗仪, 膝关节炎, 疼痛 celecoxib, bone trauma instrument, knee osteoarthritis (KOA), pain
knowledge graph	 <p><塞来昔布, 治疗, 膝关节炎> <celecoxib, treat, KOA> <膝关节炎, 临床症状, 疼痛> <KOA, clinical symptom, pain> <塞来昔布, 不良反应, 疼痛> <celecoxib, adverse drug reactions (ADRs), pain></p>
text	目的：探讨塞来昔布联合骨创伤治疗仪治疗膝关节炎疼痛的疗效。方法：将聊城市第三人民医院骨科门诊2015年1月—2017年1月收治的108例单侧早中期膝骨性关节炎患者随机分为观察组、对照组，每组54例。观察组口服塞来昔布加上应用骨创伤治疗仪；对照组口服塞来昔布。比较两组患者临床疗效。结果：末次随访时，观察组患者疼痛评分均低于对照组（ $t=3.21, p=0.00$ ），观察组患者膝关节功能优于对照组（ $t=3.74, p=0.00$ ），观察组总有效率88.89%高于对照组的77.78%（ $\chi^2=4.70, p=0.03$ ）。结论：塞来昔布联合骨创伤治疗仪临床疗效明显，有效降低疼痛评分，能增进膝关节功能改善，值得临床应用。 Objective: Explore the therapeutic effect of celecoxib combined with bone trauma treatment instrument on knee arthritis pain. Methods: 108 patients with unilateral early and mid-stage knee osteoarthritis treated in the orthopedic clinic of liaocheng third people's hospital from January 2015 to January 2017 were randomly divided into observation group and contrast group, with 54 cases in each group. The observation group takes celecoxib orally with the application of a bone trauma treatment device; The contrast group just takes celecoxib orally. The clinical effects of the two groups were compared. Results: At the last follow-up, the pain score of the observation group was lower than that of the contrast group ($t=3.21, p=0.00$). The knee function of the observation group was better than that of the contrast group ($t=3.74, p=0.00$). The total effective rate in the observation group was 88.89% higher than 77.78% in the contrast group ($\chi^2=4.70, p=0.03$). Conclusion: Celecoxib combined with bone trauma treatment device has obvious clinical effect, effectively reduces pain score, can improve knee function, and is worthy of clinical application.

Figure 1: An example of the medical topic-to-text task.

Intuitively, the medical knowledge graph (KG) is essential to guarantee the correctness of generated text, especially for high professional domains. However, most of the recent works don't make full use of medical knowledge graph (MKG). Lee et al. (2018) adopt an encoder-decoder model to generate free texts in electronic health records and Guan et al. (2018) propose a GAN-based framework trained by the reinforce algorithm to generate synthetic EMR text, both of which don't utilize the external medical knowledge. Lee et al. (2019) incorporate medical concept embedding into the sequence-to-sequence

[†] Work done when Youcheng Pan was an intern at Peng Cheng Laboratory.

* Corresponding author

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

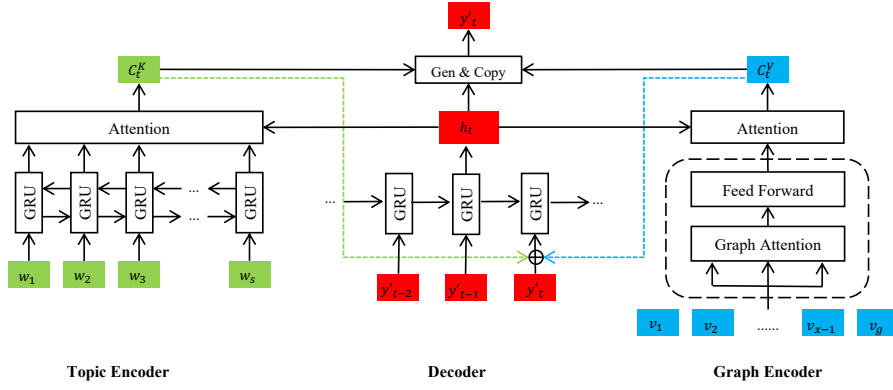


Figure 2: The overview of the MedWriter model.

model, which is pretrained by leveraging the medical concept unique identifiers from the UMLS, to improve the quality of generated clinical text. However, they view each triplet as an instance and adopt the Skip-gram algorithm (Mikolov et al., 2013) to acquire the pretrained concept embedding, which ignores the relationships between medical entities.

In this paper, we focus on the knowledge-aware medical text generation, which is a topic-to-text task. We firstly collect a medical literature dataset from medical journals that contains more than 50,000 topic-text pairs. Each of them has a set of keywords describing the topic and a relevant abstract as target text. For each pair, we collect the corresponding knowledge from a large scale Chinese medical knowledge graph CMeKG¹. An example is shown in Figure 1. Then, we adapt a knowledge-aware neural generation model for this task, named MedWriter, which consists of three components: topic encoder, graph encoder and decoder. The topic encoder is used to acquire the representation of topic words, while the graph encoder exploits the specific information from the MKG. Therefore, the model combines the information of topic words with the medical knowledge. Afterwards, we use the decoder with copy mechanism (Gu et al., 2016) to generate medical text. Experimental results demonstrate incorporating knowledge graph into generation model can improve the quality of the generated text and has robust superiority over the competitor methods.

2 Method

Given a set of topic words $K = \{w_1, w_2, \dots, w_s\}$, and a knowledge graph represented as a set of triples, i.e., $G = \{g_1, g_2, g_3, \dots\}$, where each triple g_i is comprised of $\langle s_i, p_i, o_i \rangle$ denoting subject, predicate and object respectively, our goal is to generate a natural language text $Y = \{y_1, y_2, y_3, \dots\}$, which is required to be relevant to the topic, grammatically correct and informative.

2.1 Topic Encoder

We first convert each keyword into word embedding representation $e(w_i)$ by a matrix $M \in \mathbb{R}^{l \times d}$, where l denotes the size of vocabulary and d denotes the dimension of word embedding. Then a bidirectional GRU (Cho et al., 2014) is employed to transform the keywords into a distributed representation:

$$\vec{h}_t = \overrightarrow{\text{GRU}}(\vec{h}_{t-1}, e(w_t)), \quad \overleftarrow{h}_t = \overleftarrow{\text{GRU}}(\overleftarrow{h}_{t+1}, e(w_t)), \quad r_{w_t} = [\vec{h}_t; \overleftarrow{h}_t] \quad (1)$$

where $[\cdot]$ denotes the concatenation operation; $e(w_t)$ denotes the word embedding of the t -th keyword. The last hidden states of the forward and backward GRU network are concatenated as the entire keywords representation $\vec{r}_K = [\vec{h}_s; \overleftarrow{h}_0]$.

2.2 Graph Encoder

For each knowledge graph, as the previous graph-based works did, we first perform a Levi graph transformation (Beck et al., 2018), where each labeled edge in G is replaced by two unlabeled edges, and add

¹<http://cmekg.pcl.ac.cn/>

reverse and self-loop edges to the Levi graph. For instance, given a triple $\langle s, p, o \rangle$, after transformation, we obtain $\langle s, \rightarrow, p \rangle$, $\langle p, \rightarrow, o \rangle$, $\langle o, \rightarrow, p' \rangle$, $\langle p', \rightarrow, s \rangle$ and their self-loop connections, where p' is the reverse edge of p . In this way, both the entities and relations can be viewed as vertices without losing any information. Besides, a global vertex is added to connect all entity vertices in order to aggregate the information between disconnected parts of graph. Thus, the original knowledge graph can be represented as a unlabeled graph $G' = \{V, E\}$, where $V = \{v_1, v_2, \dots, v_{x-1}, v_g\}$ is a list of entities, relations and global node v_g , and E is an adjacent matrix $M \in \mathbb{R}^{x \times x}$ which describes the connections, where x is the total number of the vertices contained in V . If v_j is a neighbour of v_i , then $E(v_i, v_j) = E(v_j, v_i) = 1$, otherwise 0.

The graph encoder is composed of a stack of several identical layers similar to (Vaswani et al., 2017), each of which has a multi-head attention sub-layer followed by a feed-forward network sub-layer. Each sub-layer is equipped with a residual connections (He et al., 2016) and a layer normalization (Ba et al., 2016). With the same operation in 2.1, the vertices are converted to an embedding representation $e(v_i)$.

Following a similar procedure to (Koncel-Kedziorski et al., 2019), for each vertex v_i , in order to obtain the contextual representation, we adopt multi-head attention mechanism to attend over the other vertices adjacent to v_i in G' . It linearly projects the inputs of attention several times with different parameters respectively. All the inputs of attention function come from V , and then the multi-head self-attention can be calculated as:

$$MulHeadAtt(V) = [head_1; head_2; \dots; head_n]W, \quad head_t = \{r_{v_1}^t, r_{v_2}^t, \dots, r_{v_{x-1}}^t, r_{v_g}^t\}, \quad (2)$$

$$r_{v_i}^t = \sum_{j \in N_i} \frac{\alpha_{ij}}{\sqrt{d_h}} W_1^t e(v_j), \quad \alpha_{ij} = \frac{\exp(e(v_j)^T W_2^t e(v_i))}{\sum_{k \in N_i} \exp(e(v_k)^T W_2^t e(v_i))} \quad (3)$$

where N_i denotes the neighbourhood of v_i ; n denotes the number of head; d_h denotes the dimension of each head; W , W_1^t and W_2^t are learnable parameters. Then we can obtain the final output r_V of one layer by $r_V = FFN(MulHeadAtt(V))$, where FFN is a feed-forward network which consists of two linear transformations with a ReLU activation. Since the identical layers are stacked for several times, where the output of previous layer is fed into current layer as input, we take the output of the last layer as the final encoding representation.

2.3 Decoder

We use an attention-based GRU network as the decoder initialized by the concatenation of the representations of topic and global vertex $[\overline{r_K}; r_{v_g}]$. At the t -th time step, the hidden state h_t is calculated by $h_t = GRU(h_{t-1}, e(y'_{t-1}), c_{t-1})$, where h_{t-1} is the hidden state of last step; $e(y'_{t-1})$ is the embedding of the output of last step; c_{t-1} is the context embedding in the last step. The context embedding c consists of two parts: c^K and c^V , attending over keywords and knowledge graph respectively.

$$c_{t-1} = [c_{t-1}^K; c_{t-1}^V], \quad c_{t-1}^V = \sum_{i \in V} \alpha_i W_3 r_i, \quad \alpha_i = \frac{\exp(h_{t-1}^T W_4 r_i)}{\sum_{j \in V} \exp(h_{t-1}^T W_4 r_j)} \quad (4)$$

where W_3 and W_4 are learnable parameters. The computation of c^K is similar to c^V . Meanwhile, we also adopt the copy mechanism (Gu et al., 2016) to directly select the token from keywords and knowledge graph. The probability p for copying is computed as $p = \sigma(W[h_t; c_t] + b)$. Then we can obtain the final probability distribution:

$$(1 - p) * \mathbf{P}_{gen} + p * \mathbf{P}_{copy} \quad (5)$$

where \mathbf{P}_{gen} is a probability distribution over all words in the vocabulary which is calculated by two linear neural networks with a softmax function; \mathbf{P}_{copy} is a probability distribution of copying a word from inputs based on the attention scores over the $[K; V]$

3 Experiments

	item	word	entity	relation	triple
training	40960	93374	42353	174	66515
validation	5120	22129	10105	127	10322
test	5120	22139	10122	120	10299

Table 1: The statistics of the dataset.

Method	BLEU1	BLEU2	BLEU3	ROUGE-L
Seq2Seq	26.06	13.50	8.50	22.21
GraphSeq	27.06	13.82	8.63	23.01
MedWriter	30.42	17.26	11.52	26.78

Table 2: The experimental results (%).

3.1 Dataset

In order to realize the medical text generation task, we collect a Chinese medical literature dataset from medical journals. The literature dataset contains plenty of pairs, all of which come from the medical articles published on the platform. Each pair has a set of keywords describing some topic information and an abstract which is a piece of text related to the topic. However, the original pair doesn't have corresponding knowledge graph. Thus, we draw support from CMeKG which is a large-scale Chinese Medical Knowledge Graph.

Firstly, we make a mapping between keyword and entity in CMeKG. In addition to exact matching, we also conduct fuzzy matching through calculating the similarity between them. Given a keyword, we select several candidate entities based on the inverted index we built and then utilize the WMD algorithm (Kusner et al., 2015) to compute the similarity between keyword and each candidate entity. We use a lot of medical literature to pretrain the char embedding. When calculating the similarity, we keep the entity with the highest score among the entities whose score is more than 0.7. Afterwards, given a set of entities, all pairwise entities are used for search in CMeKG and we keep the exact matched triples. Besides, we consider the fuzzy matching as a new relation and keep all the \langle keyword, fuzzy matching, entity \rangle triples. Finally, we obtain a dataset that contains more than 50,000 items. Each item has a set of topic keywords, an abstract as text and a corresponding knowledge graph derived from CMeKG. The statistics of the dataset are shown in Table 1.

3.2 Competitor Methods

In order to validate the effectiveness of incorporating knowledge graph into generation model, we compare MedWriter with two competitor methods.

The first method is an attention-based sequence-to-sequence model (Sutskever et al., 2014), which only use the topic words as input to generate text, named **Seq2Seq**.

The second method is a variant of the Seq2Seq, which utilizes not only the topic words but also the linearized knowledge graph, named **GraphSeq**. Borrowing the idea from (Konstas et al., 2017), we flatten the knowledge graph to a linear sequence according to the entity order they appear in the text. Another sequence encoder is employed to encode it.

When decoding, both of the two competitor methods are equipped with copy mechanism.

3.3 Settings

The model is trained to minimize the negative log-likelihood of the training set with the SGD optimization. The learning rate is set to 0.15. The hidden size of GRU is set to 512. The stack of graph encoder has 6 identical layers. We employ 4 parallel attention layers to perform multi-head attention. The dimension of embedding layer and the attention sub-layer are set to 512, while the intermediate dimension of linear sub-layer is set to 2048. The size of the vocabulary is truncated to 50,000. The batch size is set to 32. We train the model for 30 epochs and select the model which achieves the best performance on the validation set.

3.4 Metrics

For evaluation, we adopt BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics. BLEU is an n-gram overlapping measure which is widely adopted in the text generation task. BLEU1, BLEU2 and BLEU3 are reported. ROUGE is also a common measure to automatically determine the quality of the generated text. We report the F1 score for ROUGE-L, which measures the longest common sequence (LCS) between the reference and the candidate.

3.5 Results

As shown in Table 2, the Seq2Seq method achieves the worst performance in terms of both BLEU and ROUGE since it only uses the topic keywords. The GraphSeq method outperforms the Seq2Seq because it uses the medical knowledge graph, though the graph is viewed as a sequence consists of entities and relations, which means the knowledge graph can improve the performance a little bit with this setting. Compared to the competitor methods, MedWriter significantly improves the performance by at least +3.36 BLEU1 points and +3.77 ROUGE-L points while the GraphSeq just improve the Seq2Seq by +1.0 BLEU1 points and +0.8 ROUGE-L points, which means incorporating the knowledge graph including not only the entities and relations but also the graph structure into generation model is indeed conducive to the medical text generation task. Like the triple containing the relation between entities, besides that, the knowledge graph even contains the relation between triples. The experimental results also illustrate the point.

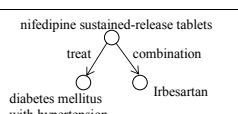
topic	硝苯地平缓释片, 厄贝沙坦, 糖尿病合并高血压 nifedipine sustained-release tablets, Irbesartan, diabetes mellitus with hypertension
knowledge graph	<p><硝苯地平缓释片, 联用, 厄贝沙坦> <nifedipine sustained-release tablets, combination, Irbesartan ></p>  <p><硝苯地平缓释片, 治疗, 糖尿病合并高血压> <nifedipine sustained-release tablets, treat, diabetes mellitus with hypertension></p>
generated text	<p>目的：探讨厄贝沙坦联合硝苯地平缓释片治疗糖尿病合并高血压的效果。方法：选取 2017 年 1 月 ~ 2019 年 1 月我院收治的糖尿病合并高血压患者 60 例作为研究对象，随机将其分为对照组和观察组，各 30 例。对照组采用硝苯地平缓释片治疗，观察组在对照组的基础上加用厄贝沙坦治疗。比较两组患者的治疗效果。结果：观察组患者的治疗总有效率（95.00%）明显高于对照组（80.00%），差异有统计学意义（$p < 0.05$）。观察组患者的治疗总有效率明显高于对照组，差异有统计学意义（$p < 0.05$）。结论：厄贝沙坦联合硝苯地平缓释片治疗糖尿病合并高血压的效果显著，值得临床推广。</p> <p>Objective: Investigate the effect of Irbesartan combined with nifedipine sustained-release tablets in the treatment of diabetic patients with hypertension. Methods: 60 patients with diabetes and hypertension treated in our hospital from January 2017 to January 19 were selected as the research object. It was divided into a contrast group and an observation group, 30 cases each. The contrast group was treated with nifedipine sustained-release tablets, and the observation group was added with nifedipine sustained-release tablets. The treatment effects of the two groups were compared. Results: The total effective rate of treatment in the observation group (95.00%) was significantly higher than that in the contrast group (80.00%), and the difference was statistically significant ($p < 0.05$). The total effective rate of treatment in the observation group was significantly higher than that in the contrast group, the difference was statistically significant ($p < 0.05$). Conclusion: Irbesartan combined with nifedipine sustained-release tablets has a significant effect on diabetic patients with hypertension, and is worthy of clinical promotion.</p>

Figure 3: An example generated by MedWriter.

The Figure 3 shows an example of the generated text by MedWriter. The generated text is of good quality on syntactic and semantic except for some repetition. And these topic words and their relations are also described in the text. It demonstrates that the MedWriter has the ability to model the knowledge graph and learn the information contained in it. Though MedWriter achieves a nice performance, but there are still many issues unsolved. Medical literature always contains a lot of medical indications and their corresponding values. If the model generates a right description of indication but a wrong value, the entire generated text may be meaningless even hazardous in the medical domain. For example, in the generated text, though 95% is higher than 80% which conforms to the description, the numerical values aren't necessarily accurate, while these values are very important and often appear in the medical text. So how to generate a right numerical value for the corresponding term is a considerable and challenging problem, we will explore it in the future research.

4 Conclusion

We use the medical knowledge graph to facilitate the medical text generation. A Chinese medical literature dataset with the corresponding knowledge graph is collected and an encoder-decoder model equipped with a graph encoder is adapted to the medical topic-to-text generation task. Experimental results demonstrate the effectiveness of incorporating knowledge graph into generation model by outperforming the competitor methods. This work is a preliminary attempt on knowledge-aware medical text generation. In the future, we plan to do more researches on applying natural language generation technology to the medical domain.

Acknowledgements

This work was jointly supported by the Natural Science Foundation of China (Grant No. 62006061, 61872113), the Strategic Emerging Industry Development Special Funds of Shenzhen (Grant No. JCYJ20190806112210067), CCF-Baidu Open Fund (Grant No. CCF-BAIDUOF2020004) and Baidu.

References

- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.
- Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. 2018. Generation of synthetic electronic medical record text. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380. IEEE.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381*.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.
- Wangjin Lee, Hyeryun Park, Jooyoung Yoon, Kyeongmo Kim, and Jinwook Choi. 2019. Clinical text generation through leveraging medical concept and relations. *arXiv preprint arXiv:1910.00861*.
- Scott H Lee. 2018. Natural language generation for electronic health records. *NPJ digital medicine*, 1(1):1–7.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July.
- Oren Melamud and Chaitanya Shivade. 2019. Towards automatic generation of shareable synthetic clinical notes using neural language models. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 35–45.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- 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 of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- 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.
- 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.
- Tianlin Zhang, Pei Quan, et al. 2018. Domain specific automatic chinese multiple-type question generation. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1967–1971. IEEE.