Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/45/20241023

A survey of text generation models

Wenhan Liang

Beijing University of Posts and Telecommunications, Haidian District, Beijing, 100876, China

liangwh2013@163.com

Abstract. In this article, I propose four model classifications to summarize the characteristics and analyze the advantages and disadvantages of text generation models that have emerged in recent years, so as to give researchers an overall overview. The models based on the decoder only use the decoder for text extraction, and its output only depends on the previous output. The models based on the encoder-decoder, on the other hand, refer to both the encoder's output and the previous prediction. I've deliberately categorized prefix models and ensemble models to highlight their differences. I also present the current state of the text generation field and compare the advantages and disadvantages of several of these models. Finally, I summarize the difficulties encountered in the field of text generation and provide a research direction for the field. In the module Challenges, I focused on the problem of scarcity regarding datasets. The current solutions are given, as well as the efforts made by relevant workers on domain-specific datasets.

Keywords: Text Generation, Decoder, Encoder.

1. Introduction

Text generation is currently a very important but challenging task in the field of natural language processing, which aims to generate readable natural language text for more representative applications, such as dialogue systems, text summarization, and machine translation.

At present, with the continuous development and popularity of neural network technology and Transformer, text generation techniques have been developed rapidly as never before in recent years and gradually applied to a number of fields. Currently, rule-based, statistical-based and deep learning-based text generation models have emerged, and have shown their unique performance and advantages in different application scenarios. There are many text generation models being produced, but there has not been a more comprehensive classification for overview. This thesis hopes to classify and summarize the models proposed in recent years, and analyze the advantages and disadvantages of each type. In this paper, I focus on four different model architectures to introduce text generation models, namely, decoder models, encoder-decoder models, prefix models and ensemble models.

The decoder-based models such as GPT2[1], CTRL, etc. are all based on the input text going through the encoder for feature extraction, and then the decoder passes the extracted features and generates the corresponding output text. The encoder-decoder based models have better context understanding ability, such as T5[2], BART [3], whose decoders use the encoder output information to generate char- acters, which is the most common type of models. Both of these models require relatively large data sets, but both have excellent performance. In particular, the T5 model can implement many natural language

DOI: 10.54254/2755-2721/45/20241023

processing tasks by fine-tuning techniques. Prefix models can better capture the complexity and long-term dependency of contextual information. This model mainly introduces mask matrix to perform random masking of some tokens. It can effectively improve the accuracy of the model for the following, and this method is also useful in encoder-decoder models. Ensemble models, on the other hand, use multiple model voting mechanisms to improve performance, but the algorithms are more complex.

2. Models

2.1. Based on the decoder

Decoder-only text generation models, also called autoregressive language models, are mostly structured as a one-way recurrent neural network (RNN) or Transformer decoder network, where each time step uses the words or characters already generated before to generate the next words or characters. The model will define a probability distribution that, given the previously generated words, predicts the probability distribution of the next words so that one of the words is selected as the predicted output to be used as input for the next time step. The decoder-based model is usually trained using maximum likelihood estimation. In the training phase, for each time step, the model compares the true next word with the predicted next word and updates the model parameters based on the error of the comparison result.

The GPT2 [1] model, unlike previous single-task datasets, uses a large corpus of text to learn the patterns of linguistic structure. Unsupervised learning was then performed with two unlabeled corpora, and then fine-tuned using an annotated task, using a decoder to predict the next character of the sequence. DialoGPT[4] adopts the same Transformer architecture as GPT-2 and pre-trains on large- scale dialog data, which allows DialoGPT to better learn the features and structure of dialog texts. A dynamic positional encoding method is used to better handle variable-length input sequences in dialog texts. A conditional layer normalization technique is used in the decoder part to introduce contextual information and generate more fluent text. PLANET[5] uses an autoregressive decoder first performs dynamic content planning by generating a latent representation (SNj) as a semantic guide, and then generates sentence words.CTRL1 controls text generation by including text labels as part of the input, and prefixes the specific content of each sequence with a description of the input type. Control instructions are added to the training data using a multilayer Transformer structure with a shallow joint training approach.

The decoder-based model can handle variable-length sequence inputs and does not require alignment operations on the target sequence; however, it is prone to bias and noise accumulation due to the inability to use the information above.

2.2. Based on the encoder and decoder

Encoder and decoder-based models are the most widely used text generation models. Encoder networks encode the input text data into a fixed-length vector, while decoder networks can use this encoded vector to generate multiple output text sequences. During training, the Decoder network uses a teacher-forcing mechanism, where the correct output of the previous moment is used as input for the current moment, so that the model can better learn the correct language rules and semantics.

Many models use the masked technique for data training, hoping to destroy information about the structure of the sequence and prevent the model from "relying" on such information. In BART [3] and MASS [6], the model blocks a word or multiple consecutive words at random, while in Pegasus [7], important sentences are blocked from the input document and generated together as an output sequence from the remaining sentences. In T5[8], a large label-free corpus is utilized, and static and dynamic word vector representations are used to encode the text. AraT5 [9] replaces URLs and user mentions with URL and USER to reconstruct the word list and implement the small language T5 model. Hierarchical Reinforcement Learning (HRL)[10] proposes a new approach that uses policy gradients to adjust the prior probability distribution of potential variables learned at the discourse level of a hierarchical variational model. The hierarchical policy network is combined with a variational self-encoder for decision making via MDP. DialogVED [11] uses a multi-layer transformer-based encoder to encode

DOI: 10.54254/2755-2721/45/20241023

dialog contexts, extends the elements of the original relative distance matrix in T5 to two-tuples, and uses n-stream self-attention to predict consecutive n tokens. at the same time, consecutive latent variables are introduced into the enhanced codec pre-training framework. AugNLG [12] combines a self-training neural retrieval model with a shot less learning NLU model to automatically create data from open text. Sort n-gram phrases based on TF-IDF scores to eliminate words that are too general.

This model can better solve the problem of sentence length and multimodality, and can determine the meaning of sentences in the context more accurately. However, it takes longer time to train and is prone to overfitting. Moreover, it requires a high vocabulary and the generated results are difficult to analyze.

2.3. Prefix models

In the process of generating text, a method of prefixing the target generation sequence is used. This prefixing method can control the generation direction of the generated sequence and make the generated sequence more consistent with the rule constraints of the task.

KM-BART [13] uses a convolutional neural network to extract visual embeddings by special tokens to inform models with different input patterns, and uses an autoregressive one-way decoder to replace the general visual embeddings with special tokens as inputs. UNILM [14] uses the transformer structure to randomly mask the input sequence and let the model predict the words corresponding to the mask position, and let the model learn to determine whether the two input texts are consecutive contexts during the pre-training process.UNILMv2 [15] adopts the "no-prompt" strategy, replacing the first two tokens in the predicted position with [SEP], instead of replacing the first token with [MASK] as in UNILM, which improves the model generalization performance. GLM[16] uses a dynamic masking technique to dynamically mask the text during the prediction process, thus reducing data sparsity and improving the accuracy and generalization performance of the model.

Prefix models use a forward computation approach that is more computationally efficient when dealing with long sequences. It also uses fine-tuning techniques to learn multiple natural language processing tasks simultaneously. However, this model has higher data requirements and the number of parameters of the model is large.

2.4. Based on the ensemble model

The integrated model is a more comprehensive generation model that votes on the results generated by multiple models and selects the result with the most votes as the final output. It is also possible to integrate multiple generative models as sub-models into one neural network model, where each generative model is considered as a sub-model and the integrated model is responsible for integrating and fusing their generative results.

ProphetChat [17] uses the simulation of the inference phase for future augmented response generation. The computed weights sum the output probability distributions of the two models for each step to select k responses, and for each response n possible features are given and the best k features are selected as the final values of the selector. PLATO [18], on the other hand, uses a latent behavior identification task to identify the probability values corresponding to target responses in a given context and training data, and to compute the posterior probability distribution.

Due to the use of multiple models for computation, the integrated model possesses strong accuracy and stability, and improves the generalization ability of the model. However, it is not yet a mainstream method because of the large computational effort and the difficulty in controlling the correlation between submodels.

3. Conclusion

In this article, I give an overview of the text generation models that have emerged in recent years. I introduce four categories of text generation models: decoder models, encoder-decoder, models, prefix models and ensemble models. Most of the models are based on the transformer, which can be adapted to different tasks, while the prefix models use the mask matrix to increase the generalization ability of

DOI: 10.54254/2755-2721/45/20241023

the model, so that it can predict the meaning of the following more accurately. The encoder-decoder models have excellent performance, and the fine-tuning technique based on the T5 model can be used for almost any text generalization task.

However, the training dataset is not sufficient. Language models require large amounts of textual data for training, but many current datasets do not cover a large number of domains. For this reason, GPT2 proposed the WebText [1] dataset and Google proposed the C42 dataset, but there is still a lack of high-quality datasets. Another challenge is the underdeveloped domain-specific text generation. In some fields, such as medicine and law, specialized knowledge is required. Knowledge augmentation can be used to address this issue. Graph2Seq [19] uses graph neural networks for knowledge graph interpretation. Chen Xing[20] et al. used a specific penalty mechanism for topic word selection and importance determination with the help of topic words in generative topic models.

Text generation is a very important area of natural language processing. It is foreseeable that all the above challenges will be solved in the future and new models will be generated. But the basic ideas are derived from the extensions of the above four models to improve the performance by more means.

References

[1] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya S utskever. Language models are unsupervised multitask learners. 2019.2 https://paperswith code.com/dataset/c4

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(1), jan 2020.

[3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. sequence pre- training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[4] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT: Large-scale generative pre-training for conversational response generation. In Proceedings of the 58th Annual Meeting of the Association for Compu- tational Linguistics: System Demonstrations, pages 270 – 278, Online, July 2020. Association for Computational Linguistics.

[5] Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. PLANET: Dynamic content planning in autoregressive transformers for long-form text generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2288–2305, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[6] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In International Conference on Machine Learning, 2019.

[7] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. ArXiv, abs/1912.08777, 2019.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020.

[9] El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. AraT5: Text- to-text transformers for Arabic language generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 628 – 647, Dublin, Ireland, May 2022. Association for Computational Linguistics.

DOI: 10.54254/2755-2721/45/20241023

[10] Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, and Rosalind W. Picard. Hierarchical reinforcement learning for open-domain dialog. In AAAI Conference on Artificial Intelligence, 2019.

[11] Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. DialogVED: A pre-trained latent variable encoder-decoder model for dialog response generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4852 – 4864, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[12] Xinnuo Xu, Guoyin Wang, Young-Bum Kim, and Sungjin Lee. AugNLG: Few-shot natural language generation using self-trained data augmentation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1183–1195, Online, August 2021. Association for Computational Linguistics.

[13] Yiran Xing, Zai Shi, Zhao Meng, Gerhard Lakemeyer, Yunpu Ma, and Roger Wattenhofer. KM-BART: Knowledge enhanced multimodal BART for visual commonsense generation. In Proceed- ings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 525 – 535, Online, August 2021. Association for Computational Linguistics.

[14] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified Language Model Pre-Training for Natural Language Understanding and Generation. Curran Associates Inc., Red Hook, NY, USA, 2019.

[15] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In International Conference on Machine Learning, 2020.

[16] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320 – 335, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[17] Chang Liu, Xu Tan, Chongyang Tao, Zhenxin Fu, Dongyan Zhao, Tie-Yan Liu, and Rui Yan. ProphetChat: Enhancing dialogue generation with simulation of future conversation. In Proceed- ings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 962 – 973, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[18] Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. PLATO: Pre-trained dialogue generation model with discrete latent variable. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 85– 96, Online, July 2020. Association for Computational Linguistics.

[19] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In Proceedings of the 56th Annual Meeting of the Association for Compu- tational Linguistics (Volume 1: Long Papers), pages 273–283, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[20] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic aware neural response generation. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, page 3351–3357. AAAI Press, 2017.