Investigating Natural Language Processing algorithms to generate automatic and adaptive learning materials

Chiranjeev Keshav Nathoo
Computer Science Honours
University of Cape Town
Cape Town, South Africa
NTHCHI002@myuct.ac.za

ABSTRACT

In the recent years, the use Natural Language Processing (NLP) for text generation has been an area of great interest. Researchers have been invested in how it can be used to enhance the learning process among students. This is where the idea for adaptive learning materials immerge. The knowledge level of the learner is assessed through a series of questions. To fill the gaps in his knowledge, appropriate content is retrieved from structured knowledge bases. Expressing this content in a human-understandable manner is where NLP is used. This literature review aims to explore NLP and the works that have been done in this field to gain an insight into the different ways that gathered contents adapted to the learner can be rearranged and expressed into a natural language. This is the task explored in this paper. These methods will also be compared and contrasted against each other to find out how each of them can fulfil the purpose of this project and discuss whether a combination of them can provide a better solution for creating documents from a structured knowledge.

CCS CONCEPTS

• **Computing Methodologies** ~ Natural Language Generation ~ Neural networks

KEYWORDS

Natural Language Processing, Natural Language Generation, Automatic Question Generation, Ontology, Machine Learning

1 Introduction

E-learning and adaptive learning materials has gained a lot of popularity as technology improved over the years and has become more accessible to the populace. As a result, developing an application that will assess the knowledge level of its user, in this case, a learner, through the means of a series of questions and provide the former with tailored learning materials is expected to greatly improve the quality of education and make self-learning much more beneficial in the eyes of learners [1]. It will potentially also reduce the workload of teachers and the time and resources they spend to make adapted learning materials manually.

This literature review aims to investigate how content deemed suitable for the learner can be grouped together and presented in a human-readable manner that is, using a natural language. This is covered mostly by Natural Language Generation (NLG), a subset of NLP. NLG is defined as the field that has to do with the generation of human-understandable text from some underlying non-linguistic representation of data [17]. NLG has had a lot of real-world applications to handle presentation of data in the recent years [17], however very few of them have been useful on a large scale, despite being theoretically proven to be effective. There is research in the field that shows that the use of neural network-based NLG techniques for text generation can been used to produce semantically and syntactically accurate content and with superior quality compared to basic templated methods [21]. However, template-based methods for text generation are also relevant here because they require much less computational power, have been tested for a much longer period than neural network-based NLG methods and still produce useable content [16].

This paper will cover three main areas that are important to our research objective. Firstly, a brief introduction on text expansion and what it entails. This will be followed by template-based methods and some of their applications. Then, an exploration of some relevant and state of the art NLG techniques. Finally, the mentioned methods will be discussed and a list of methods that can be potentially used to fulfil the objective of the task being investigated in this paper will be concluded.

2 Text Expansion

This is one of the uses of NLG and the one that we are interested with in this review. It creates texts that are a richer representation of the short texts provided to the system based on a structured knowledge base [6]. It can be split into two sections namely, short-text expansion and topic-to-essay generation (TEG). We are more specifically interested in the latter as it is very closely related to the task being investigated.

Essay generation takes topic words as input and outputs a paragraph which must maintain topic integrity and relevance [8].

Methods for text expansion can be template-based or neural network-based. In our case, if gathered contents stored in a structured knowledge are considered as topics, they can be used to generate the related document. One issue with TEG is that most of the time, the input information is extremely insufficient which makes it hard to be consistent with the topic [25].

3 Template-based methods

Reiter *et al.* [17] defines a template system as one that represent sentences as boilerplate text and parameters that need to be inserted into the boilerplate without any intermediate representations. This can be argued to be one of the advantages of templates as they require fairly low computational power. Another advantage is that since templates can be described as a deterministic approach to text generation, good linguistic rules can be specified within the template [5]. To make sure that the input parameters are correctly placed in the template's slots, template rules need to be defined. If there are not a lot of inputs that require a rule to categorise and place them then using a template does not require a lot of effort.

Since templates are generally considered to be inflexible and hard to maintain [17], a lot of effort has been put in to try and generalise them over the years. Mcroy *et al.* [13] proposes an approach to augment templates by allowing template slots to be filled by other declared templates and the ability to recover from omissions and errors by providing some default values. Their method can specify content from a knowledge base which reduces the amount of linguistic knowledge that needs to be embedded in the template. Their method is a good example of how templates can be used for our task.

4 NLG Architectures

These are the architectures that are generally used in NLG. A single or a combination of them can be used as the basis of an NLG system. They allow the NLG system to be more automated and less restrictive than the traditional template-based approaches for text generation [16].

4.1 Recurrent Neural Network

This is a class of artificial neural network which when given a sequence of data as input, can summarise and store that in its hidden state which is recursively updated. New samples can then be generated by the network based on a probability distribution specified by the hidden state [18]. Tang *et al.*[18] used RNNs to create two fake review generators. In their experiment, more than 50% of the generated fake reviews were misclassified by human judges. This shows the effectiveness of RNNs. Indurthi *et al.* [11] propose a way to use a knowledge base to generate question and answers. Since our task also involves creating sentences from a knowledge base, it was interesting to study how they do this since questions are also a type of sentence. Using the knowledge base, they create an ordered set of keywords which is used as input to a bi-directional RNN, essentially treating this as a sequence-to-sequence problem. Their results show that although performance is better overall when compared to template-based method, there are still some erroneous sentences that are produced.

RNNs traditionally suffer from an issue called vanishing gradient when dealing with long sequence of inputs which is when a neural network cannot back-propagate information because of the number of layers and the loss function. Zaremba *et al.*[28] show how using a long-short term memory (LSTM) [30] unit with the RNN can mitigate this problem to some extent.

4.2 Memory Network

This is type of neural network with a memory block. Weston *et al.* [22] states that a memory network consists of a memory, m which usually a data structure like an array, and four other components. An input feature map which converts input into an internal representation. Generalisation which is updating old memories as new input comes in. This prevents the model to get overtrained on the same inputs. An output feature map which produces the output and changes based on the input and data in memory and finally a response which is a secondary input to specify in which format the output must be converted into.

Chen *et al.* [3] experimented with using two memory networks for image and text respectively on top of their existing transformer-based encoder-decoder to improve radiology report generation. They found that incorporating the memory networks allowed better analysis of the provided X-ray images and produced better reports than the previous iteration of the system. Having that additional memory for processing data improved the system's performance.

4.3 Generative Adversarial Network

As defined by Goodfellow *et al.* [9], the goal of a generative adversarial network (GAN) is to estimate the probability distribution from a set of training examples and use that estimate to generate more examples. GANs comprise a generator and a discriminator and uses game theory to train themselves. The generator uses the input data distribution to try to produce fake samples. The distributor tries to identify whether the sample came from the actual input, or the ones created by the generator. This training process continues until the distributor cannot distinguish between fake and real input samples [6]. The generator and discriminator are both neural networks. The type of neural network chosen for each of them depends on the use of the system.

Zhang *et al.* [29] proposed a framework, TextGAN, which uses GAN to generate realistic-looking sentences. A RNN was used as a generator and a Convolutional Neural Network (CNN) as a discriminator. They compare TextGAN to other existing GAN based text generators such as SeqGAN [27] and a baseline autocoder using encoder-decoder architecture among others. They concluded that TextGAN had superior performance based on the results of their experiments. This shows that GANs can outperform RNNs in the document generation task.

4.4 Attention Mechanism

An attention mechanism iteratively processes its input by selecting relevant content at each step [4]. Vaswani *et al.* [19] describes it as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. It is inspired by how humans selectively focus on distinctive parts while processing large amount of data. Attention mechanisms can be modified while keeping its core principle intact to better adapt them to specific tasks [14]. Niu *et al.* [14] provides a great explanation on which aspects of an attention mechanism can be altered to make it better suit its purpose. Figure 1 provides an overview.

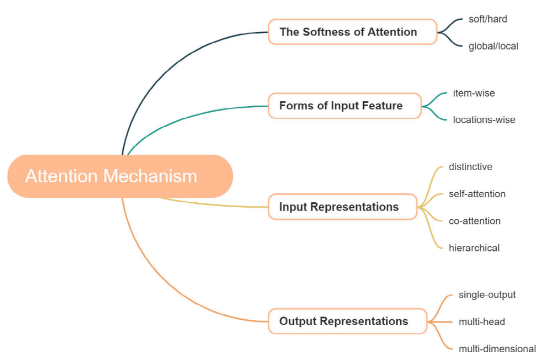

Figure 1: Several typical approaches to attention mechanisms [14]

Chorowski *et al.* [4] used attention mechanisms with some improvements, coupled with a recurrent neural network for speech recognition. Yin *et al.* [26] used attention mechanisms with a convolutional neural network to model sentence pairs. Both mentioned situations dealt with large datasets as input. The attention mechanism allowed the network to focus on the important parts of the input and not get swamped.

Self-attention deserves some extra attention since it is what is used to create transformers, currently state of the art. It is an attention mechanism that considers multiple points in a sequence and creates a representation of the sequence relating those points together [19]. This allows long distance dependencies, which basically means how words are related together, to be captured in the text, making the output sentence coherent.

4.5 Pre-trained Model

Pre-training can be defined as a way to gain a good representation from large unlabeled datasets by training a model on them and then using that representation for other tasks. This is a way to circumvent one of the challenges of most NLP tasks which is building large-scaled labelled datasets. As pre-training provides a head start to the model using the refined representation, it also allows better generalization and faster convergence to target action [15]. Erhan *et al.* [7] states that it also prevents overfitting on small datasets. The transformer is one example that works well with pre-training since it usually deals with large datasets.

4.6 Transformer

As shown in the above section, attention mechanisms are very useful when paired with neural networks and this has been the norm for a long time. Vaswani *et al.* [19] proposed the transformer model as an improvement over the previously state of the art methods. The transformer uses only attention mechanisms while following an encoder-decoder architecture where the encoder processes and the input and feeds its output to the decoder which uses the encoder's representation of the input to produce words. In the transformer, attention is known as multi-head because there are several attention layers stacked in parallel with different linear transformations of the same input [2]. The transformer is also a good example of a fully connected self-attention model This allows the transformer to process the input all at once. Wolf *et al.* [24] states that parallelisation is easier to do with transformers which allows training on larger datasets. This allows for the building of more robust and effective language models.

Guo *et al.* [10] states that to its heavy structure, the transformer usually requires a large dataset and tends to overfit on modestly sized datasets. They also propose an improvement over the original transformer model known as the star transformer which performs better for small datasets. The transformer model offers improvements over neural networks however its use must be justified as it is more complex.

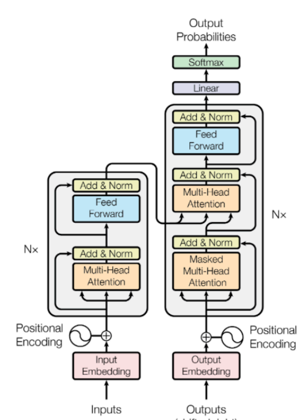

Figure 2: The transformer architecture [19]

Koncel-Kedziorski *et al.* [12] introduced a graph transforming encoder that when given the title of a scientific article and a knowledge graph, can generate a relevant abstract. They compared their solution mainly to the Graph Attention Network of velickovic *et al.* [20] because of their similarities. Their results show that using knowledge as graphs with a transformer for text generation improves performance. Figure 3 an extract from their experiment results.

Title	Block and Group Regularized Sparse Modeling for Dictionary Learning
Knowledge	(dictionary learning, CONJUNCTION, sparse coding) ; (optimization problems, USED-FOR, dictionary learning) ; (optimization problems, USED-FOR, sparse coding)…
GraphWriter	Sparse representations have recently been shown to be effective in many optimization problems. However, existing dictionary learning methods are limited in the number of dictionary blocks, which can be expensive to obtain. In this paper, we propose a novel approach to dictionary learning based on sparse coding …
GAT	In this paper, we consider the problem of dictionary learning in well-known datasets. In particular, we consider the problem of dictionary learning, where the goal is to find a set of dictionary blocks that maximize the likelihood of a given set of dictionary blocks …
EntityWriter	We propose a novel dictionary learning framework for reconstructed block/group sparse coding schemes. The dictionary learning framework is based on the descent, which is a block structure of the group structure …
Rewriter	This paper presents a new approach to the k-means of the algorithm. The proposed approach is based on the basis of the stationarity algorithm. The algorithm is based on the fact that the number of bits is a constant of the base of the base of the input …
Gold	This paper proposes a dictionary learning framework that combines the proposed block/group (BGSC) or reconstructed block/group (R-BGSC) sparse coding schemes with the novel Intra-block Coherence Suppression Dictionary Learning algorithm. An important and distinguishing feature of the proposed framework is that all dictionary blocks are trained simultaneously …

Figure 3: Results of the graph transformer compared to similar systems [12]

5 Discussion

As mentioned in the above section, templates can be used to fulfil our objective. The content from the knowledge base can be made to fit in the template whilst keeping any linguistic value using rules. Templates allow us to maintain a consistent sentence structure and most of the time are re-useable. However, traditional templates also have some limitations. Too much time and resource are wasted to create different templates for each use case. We need the templates that can maintain cohesion but can also adapt. To this end, Wiseman *et al.* [23] proposes a way to use RNNs to be able to change the parameters of a Hidden Semi-Markov Model (HSMM) as they say this statistical model best describes how a template functions. Each state of the HSMM is aligned with a slot in the template and retraining the RNN allows restructuring of the template while keeping the knowledge base constant. Incorporating neural networks with templates as described above is a good way to maintain balance between correctness of the sentences and generalisability of the representation. This method can be applied in our case as well where a neural network is used to generate the appropriate template and using gathered content and a knowledge base, the template can be populated.

As for NLG techniques, they are methods that have been developed only recently and as this field is gaining more and more attention, the methods are also being updated and improved. All mentioned NLG architectures require training on some dataset before they can be used for document generation since they use some form of neural network. RNNs and memory networks will be discussed together as they are very similar with the one key difference being that memory networks have a larger memory for their hidden states. Both methods can be made to accept input collected from knowledge bases by transforming the latter into a set of keywords as we have seen previously [11]. This is a useful feature since in our task we are also dealing with structured knowledge. After training the networks on our selected knowledge base, the contents will be fed in which will generate content-related sentences. Combining each sentence generated will give us an informative document which exactly what our task requires. An interesting point to note about the use of these methods is that they are often seen to be paired with LSTM units and attention mechanisms for document generation. A few setbacks regarding these methods are that the quality of sentence obtained as an output depends on the time spent in training the network. Furthermore, if the training dataset is not large enough there is always a risk of overfitting which is when the network becomes biased and does not recognise words outside those used during training.

GANs also make use of neural networks and as such inherits the characteristic issues but here the way training is done is what is interesting, and this works surprisingly well when wanting to do topic-related document generation. This method generates several sentences for a single set of content which means that the final document can be more detailed. This is an advantage that GANs have over RNNs and memory networks. The GAN model that Yang *et al.* [25] proposes is interesting to consider as it incorporates the use of memory networks within its generator. The memory holds data from a knowledge base which is used to make the sentences more elaborate and distinct, making this a prospective method to use. Figure 4 shows a sketch of their model.

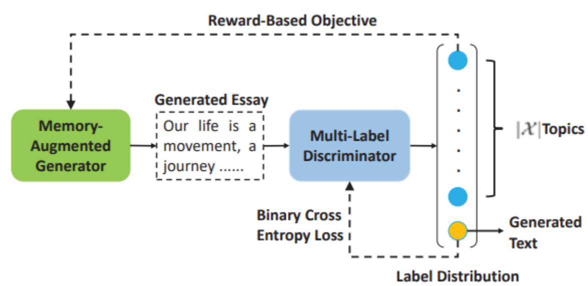

Figure 4: Sketch of GAN model [25]

Since the basis of transformers is attention mechanisms, they will be discussed together here. The use of attention mechanisms with RNNs is something that has been done prior to transformers and is still viable nowadays. They certainly provide an improvement in the quality of the document, making the system focus on the important parts of the input. Transformers, currently state of the art NLP method, can generate documents from keywords while maintaining coherence due to its multi-head attention mechanism. This mechanism is also what allows the transformer to produce an output at a lower cost and faster, compared to RNNs [19]. These are the reasons why transformers are beneficial to use for our task. However, the use of this model makes more sense where the training data is so large that parallelisation is a must to bring training time to a reasonable value. The complexity of the transformer model itself is a factor to consider.

6 Conclusions

This literature review has explored the different tools available at our disposition within the NLP field to complete the difficult task of document generation. Our task specifically requires the generation of coherent sentences based on content obtained through a knowledge base. Our research objective for this paper was to identify which different techniques can be used for the mentioned task. Each method was extensively researched to find the gap between what is currently known and the requirements of our task. We found that templates are more consistent at producing semantically and lexically correct sentences. It is also beneficial to use structured data with them as it makes it easier for template rules to classify the words into the template slots.

This paper presented that neural network and attention-based methods can process data very fast and making alterations in the output does not require a lot of effort. While transformers are the state-of-the-art method, it is still possible to use something like RNNs or GANs and obtain relatively similar document in terms of quality if long range context dependency is not a priority.

This paper has identified a gap in the knowledge about the techniques that can be used to reach our research objectives. The gap is that template-based and newer methods like neural network-based or attention-based methods are typically considered as different categories and the combination of both for a system is not something that has been widely studied. The opinion of many researchers in the NLG field is that templates are generally the inferior method in terms of reusability and diversity of sentences that it can create. However, we found that the combination of both can be used to strike a balance in terms of the malleability of the system and consistency in sentence structure. Through the use neural networks, the task of generating templates can be boosted and sentences within the document can be made to be more expressive. This is also true for the attention-based methods. To find out exactly which of the newer methods among the ones we discussed should be used in this hybrid system, experiments need to be carried out.

REFERENCES

[1] Olutayo Boyinbode, Kolawole Akintola, and Paul Olotu. 2020. Development of an ontology-based adaptive personalized e-learning system. *Applied Computer Science* 16, 4 (2020), 64–84. DOI:https://doi.org/10.23743/acs-2020-30

[2] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. 2021. An Attentive Survey of Attention Models. *ACM Trans Intell Syst Technol* 12, 5 (October 2021). DOI:https://doi.org/10.1145/3465055

[3] Zhihong Chen, Yaling Shen, Yan Song, and Xiang Wan. 2022. Cross-modal Memory Networks for Radiology Report Generation. (April 2022). Retrieved from http://arxiv.org/abs/2204.13258

[4] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. Advances in neural information processing systems, 28.

[5] Deemter, K. V., Theune, M., & Krahmer, E. (2005). Real versus template-based natural language generation: A false opposition?. Computational linguistics, 31(1), 15-24. [6] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2023. A Survey of Natural Language Generation. *ACM Comput Surv* 55, 8 (August 2023), 1–38. DOI:https://doi.org/10.1145/3554727

[7] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. *Why Does Unsupervised Pre-training Help Deep Learning?* Erhan, D., Courville, A., Bengio, Y., & Vincent, P. (2010, March). Why does unsupervised pre-training help deep learning?. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 201-208). JMLR Workshop and Conference Proceedings.

[8] Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-essay generation with neural networks. In *IJCAI International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, 4078–4084. DOI:https://doi.org/10.24963/ijcai.2018/567

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[10] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-Transformer. (February 2019). Retrieved from http://arxiv.org/abs/1902.09113

[11] Indurthi, S. R., Raghu, D., Khapra, M. M., & Joshi, S. (2017, April). Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers (pp. 376-385).

[12] Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., & Hajishirzi, H. (2019). Text generation from knowledge graphs with graph transformers. arXiv preprint arXiv:1904.02342. [13] Susan W. Mcroy, Songsak Channarukul, and Syed S. Ali. 2003. An augmented template-based approach to text realization. *Nat Lang Eng* 9, 4 (December 2003), 381–420. DOI:https://doi.org/10.1017/S1351324903003188

[14] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. 2021. A review on the attention mechanism of deep learning. *Neurocomputing* 452, (September 2021), 48–62. DOI:https://doi.org/10.1016/J.NEUCOM.2021.03.091

[15] Xi Peng Qiu, Tian Xiang Sun, Yi Ge Xu, Yun Fan Shao, Ning Dai, and Xuan Jing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences 63*, 1872–1897. DOI:https://doi.org/10.1007/s11431-020-1647-3

[16] Reiter, E. (1995). NLG vs. templates. arXiv preprint cmp-lg/9504013.

[17] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Nat Lang Eng* 3, 1 (1997), 57–87. DOI:https://doi.org/10.1017/S1351324997001502

[18] Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware Natural Language Generation with Recurrent Neural Networks. (November 2016). Retrieved from http://arxiv.org/abs/1611.09900

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[20] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[21] Raboanary, T., Wang, S., & Keet, C. M. (2022, April). Generating Answerable Questions from Ontologies for Educational Exercises. In Metadata and Semantic Research: 15th International Conference, MTSR 2021, Virtual Event, November 29–December 3, 2021, Revised Selected Papers (pp. 28-40). Cham: Springer International Publishing.

[22] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. (October 2014). Retrieved from http://arxiv.org/abs/1410.3916

[23] Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning Neural Templates for Text Generation. (August 2018). Retrieved from http://arxiv.org/abs/1808.10122

[24] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

[25] Yang, P., Li, L., Luo, F., Liu, T., & Sun, X. (2019, July). Enhancing topic-to-essay generation with external commonsense knowledge. In Proceedings of the 57th annual meeting of the association for computational linguistics (pp. 2002-2012).

[26] Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Transactions of the Association for Computational Linguistics, 4, 259-272.

[27] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017, February). Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[28] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent Neural Network Regularization. (September 2014). Retrieved from http://arxiv.org/abs/1409.2329

[29] Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., & Carin, L. (2017, July). Adversarial feature matching for text generation. In International Conference on Machine Learning (pp. 4006-4015). PMLR.

[30] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.