**A Comprehensive Review of Text Generation: From NLP to Hybrid Mechanisms**

**Abstract**

The natural language processing (NLP) field is facing a significant challenge in text generation, which is considered more complex than text understanding. The rapid expansion of electronic communication between people has made research in text generation essential. Websites across different domains now aim to respond to users using natural language. This study classifies text generation based on two principles (the level of generation and the technique used). This classification offers a comprehensive view of how text generation has developed and how different methods contribute to generating coherent and contextual text. The study recognizes deep learning as the principal approach in text generation and recommends that transforming deep learning models to include self-attention mechanisms and knowledge understanding is a promising direction for future research.

**Keywords**: Natural language processing, Human language generation, Text generation, Linguistic grammar techniques, Machine and Deep learning techniques.

**Introduction**This article clarifies the differences between three key concepts in artificial intelligence (AI): NLP, NLU, and NLG. NLP, or natural language processing, is the branch of AI focused on human language, providing techniques to understand and generate it. Within NLP, there are two main sub-fields: NLU and NLG. NLU, or natural language understanding, involves algorithms and techniques to analyze human language in various forms, including text and speech. NLG, or natural language generation, encompasses algorithms and techniques that enable the generation of human language, again applicable to both text and speech [1].

Nowadays human language generation has become a necessity in many domains. It is because of the world electronic revolution. Population searches answers to their questions trying to retrieve required information across the world wide web. So, there is no other choice, then online websites that provide applications that are available for user-direct connection; these must provide the ability of human language generation.

This article is dedicated to text generation which is a sub-branch of NLP. Text is a series of characters written in a human language. Text generation employs knowledge in artificial intelligence and computational linguistics to the automatic generation of natural language text, which satisfies certain communicative requirements [2][3].

Many applications in NLP require text generation such as; human-machine dialog, Chatbots, questioners, automatically generating abstractive summaries, machine translation, explanation generation, data-to-text generation, image-to-text generation, discourse production, text augmentation, creative writing, image caption generation, code generation, keyword-to-sentence generation, stories generation, news generation, text retelling, product title generation, and product reviews generation [4, 5, 6, 7, 8].

1. **Challenges in text generation**

Automatic text generation as a part of NLP provides many benefits, but it suffers from the problems and limitations of NLP as follows [9].

Contextual words and phrases are a big challenge to automatic text generation. The context of a sentence determines the meaning of a word because a word may have many different meanings. Such a problem is apparent in languages such as Arabic and English. So automatic text generation applications must have contextual sentence knowledge [10].

Synonym words are a challenge to automatic text generation. Many words may express the same idea, but not all of them convey the same meaning. Some synonyms may provide levels of the same meaning. Some synonyms may be used for things such as little, small, minute, and tiny. So automatic text generation applications must be provided with synonym knowledge [11].

Ambiguity is where sentences or phrases have more than one possible interpretation. It is achieved when a word in a human language has more than one lexicon classification. That is called lexical ambiguity, which could cause semantic ambiguity and syntactic ambiguity. Sometimes, such confusion requires the context of the text, while P.O.S., which stands for part of speech, can sometimes solve the problem.

Vast challenges in automatic text generation applications that provide automatic learning from training or automatic knowledge acquisition as written text, such as Irony, misspelling, Colloquialisms, and slang. Words or phrases may be negative or positive but connote the opposite; that is what is called irony. Misspelled words can be handled by spellcheckers and autocorrect applications. Colloquialisms include informal words, expressions, phrases, and idioms. All these present a vast challenge because people use slang dialects, not standard formal languages [12].

Domain-specific language means that some domains of work use very different languages. That requires building their models. Such tasks suffer from limitations because of their need for human intervention.

Limitations of automatic text generation-based machine learning techniques and deep learning techniques are focused on time-consuming, insufficiently labeled data, the lack of training data, the exposure bias problem, reward sparsity, and mode collapse. Recurrent Neural Networks (RNNs) technique can solve many problems of limitations but cannot run well with long text [13, 14].

1. **Classification of text generation**

This research classifies text generation into various categories, the most common categories are based on the level of the words produced and basic approaches [15], as shown in Figure 1.

**3.1 Classification based on word level**

Classification text generation tasks are based on the level of words produced, which varies from single words to multiple paragraphs [16].

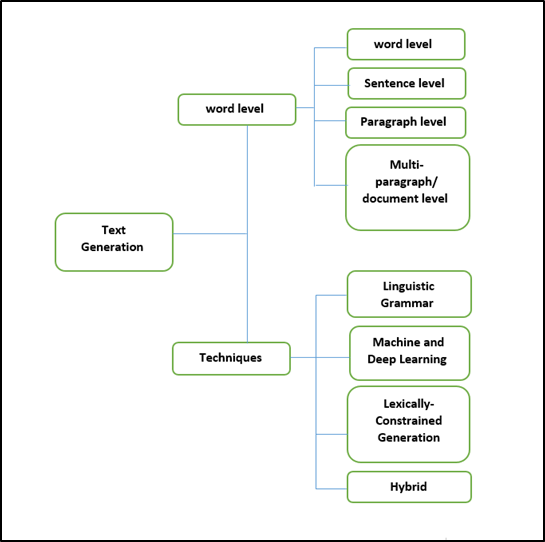
**1. Word level (next word prediction):** Utilized in text correction and intelligent keyboards, the Cloze Task, or Masked Language Modeling, functions similarly to BERT tasks where certain words are substituted with tags for prediction.

**2. Sentence level** utilized insentence completion, paraphrasing, and summarization.

**3. Paragraph level** utilized in paragraph paraphrasing, summarization**,** and generating paragraphs from information

**4. Multi-paragraph/ document level** utilized in the article, story, and report generation

The higher the level of text generation, from single words to multiple paragraphs, the more complex the task becomes, making it increasingly challenging to maintain coherence and consistency.



Classification of text generation Fig 1:

**3.2 Classification based on techniques**

Classification text generation based on automatic text generation techniques. Initially, the pure linguistic grammar mechanisms have been depicted, but because these techniques require much human intervention as well as high programming skills; researchers were directed towards machine learning techniques. Especially after providing the required datasets. Lastly, deep learning techniques recorded a big success in this domain. Today researchers presented some hybrid systems that integrate linguistic grammar and learning techniques. In addition to all mentioned mechanisms, there is another mechanism that provides the ability to specify constraints on the generated natural language text; what is known as, lexically-constrained generation [17, 18].

The following four sub-sections show each one of the automatic text generation mechanisms in detail. Then, after reviewing the various techniques; a comparison will be done between these techniques based on the applied approach, performance, ease of implementation, and the extent of human intervention required. Table 1 compares these techniques according to the key criteria discussed in the study.

1. **Linguistic Grammar Techniques**

Before the current survey period, automatic text generation was achieved by applying specific conditions to a database. In the context of artificial intelligence, a database serves as a repository for collected facts, and many related facts collectively represent what is known as knowledge. The conditions are rules applied to the input knowledge. Predicate calculus is an effective method for establishing these rules. The commonly used automatic text generation techniques based on linguistic grammar include rule-based systems, class-based systems, and statistical models.

Those techniques excelled at being well-behaved and explanatory. On the other hand; such technologies require invisible amounts of heavy hand engineering to collect and repair the knowledge and finally apply the required conditions. In addition to the pre-human intervention; those techniques required high programming skills. Those techniques made useful of linguistic grammar in building their approaches and making conclusions aiming to generate human language text. The generated text may be obtained from conjunct mini-concluded texts using conjunction tools of the required human language [19].

Gkatzia, et.al. 2017 showed that using natural language generation enhanced decision-making under uncertainty compared with the graphical-based representation method. They got 24% better decision-making using natural language generation. Then they got 44% better after merging the natural language generation and the graphical-based [20].

Qiuyun Zhang, et. al. 2019 said in their survey that the first automatic text generator was template-based. It converted data to text by providing predefined template texts containing blanks, then mining the suitable data and filling in the blanks. It built the first automatic report news which told about the California earthquake that occurred near Beverly Hills in 2014 [7].

Armin Bauer, et.al. 2015 presented a system for generating natural language text in multiple languages. The system was a rule-based text generator that had the advantage of multiple layers of reasoning. They are interested in the salesman role in e-commerce. They proposed a method to deliver all wanted information to the customer in the form of natural language text. They generated the text from the experience and the world knowledge. They proved that a rule-based text generator is better than a machine-based one in the recommending domain. They tried to generate sentences that talked about the product features as well as some comparisons with other products [21].

Emma Manning 2019 presented a new approach for generating English text from abstract meaning representation (AMR). They proved that the rule-based approach is better than neural and statistical approaches in the domain of AMR. They showed that a rule-based approach can work successfully although with limited amounts of AMR-annotated data, while other approaches require huge amounts [22].

1. **Machine and Deep Learning Techniques**

Natural text generation tasks are challenging to address, but deep learning can effectively achieve state-of-the-art results and tackle many issues in this domain. Common deep neural network techniques for natural language generation include recurrent neural networks, sequence-to-sequence architectures, convolutional neural networks, reinforcement learning, and transformers [23, 24]

Heng Wang, et. al. 2018 proposed a model for realistic text generation. Their model, VGAN; at which the generative adversarial net (GAN) is composed of merging recurrent neural network and vibrational auto-encoder (VAE). The produced model is a convolutional neural network. They considered their model as the outperform of all the others at that time [25].

Zhan Shi, et. al. 2018 addressed the two problems of reward sparsity and mode collapse. They employed inverse reinforcement learning for text generation. Firstly, they applied Markov decision process. Secondly, they formulated each text sequence in the training dataset by experts from the distribution. They learned the reward function, which explained the expert behavior. Their method has two advantages. The reward function can introduce more density reward signals. Also, the generation policy aids in generating more diversified texts [17].

Raheel Qader, et. al. 2019 presented a scheme for semi-supervised deep learning. Their scheme can learn from non-annotated as well as annotated data. They applied sequence-to-sequence models. Their approach achieved experimentally very competitive results without any need for preprocessing or re-scoring operations [26].

Markus Bayer, et. al. 2022 suggested that in machine learning applications; developing the training data may be more difficult than choosing and modeling the classifiers. They said that data augmentation methods were developed to improve classifiers via artificially created training data. They presented an approach for short as well as long text generation. Their approach used data augmentation methods that focused on deep learning algorithms. Their approach has two methods. The context-conditional method for long texts. And the context-independent method for short texts [27].

Although the sweeping empirical success of deep learning models in the domain of text generation; they are still poorly understood and may be poorly behaved at many times. Therefore, hybrid mechanisms that integrated deep learning with the consideration of knowledge understanding and syntax guidance became a necessity [28].

1. **Lexically-Constrained Generation**

The lexically-constrained generation is important for many applications that require constraints on its generated text. Those constraints can be soft such as contextualizing. Or it can be hard such as the existence of a specific keyword. The common approaches used were beam-search-based methods and stochastic searching methods. It could be applied to any one of the above mechanisms. It could be seen as an optimizing approaches that control the dataset-searching and then improve the results [29].

Peter Anderson, et. al. 2017 used a flexible approach. They improved the existing deep captioning architectures. They made useful image taggers an advantage at test time without re-training. They used a constrained beam search that forced the inclusion of selected tag words. They fixed pre-trained word embedding to facilitate vocabulary expansion to the previously unseen tag words [30].

Lei Sha 2020 solved the lexically-constrained generation via unsupervised gradient-guided optimization. His differentiable objective function used the gradient to determine which position should be changed. His method was free of parallel data training and it was flexible, so it could be used for any pre-trained generation model. He applied his method for keyword-to-sentence generation. He applied his method for unsupervised paraphrase generation, too [18].

Lianhui Qin, et. al. 2022 presented energy-based constrained decoding via Langevin dynamics. Their approach was a decoding framework that unifies constrained generation as specified constraints through an energy function. Their approach performed efficient differentiable reasoning over the provided constraints via gradient-based sampling. Their approach was flexible enough to be applied to left-to-right language models directly without needing any task-specific fine-tuning [31].

1. **Hybrid mechanisms for text generation**

Nowadays, pre-trained language models (PLMs) have become the star mechanism in natural text generation. PLMs are neural generation models based on pre-trained language models. The idea of PLMs is to provide a huge amount of unsupervised trained corpora to train the models, and then fine-tune the models in downstream supervised tasks. Then PLMs have been enforced by transformers and higher computational power. PLMs go towards deeper architectures as shown in Bert and Open AL GPT [32, 33].

Lewis M., et. al. 2019 proposed a de-nosing auto-encoder for pre-training seq-to-seq models. They called their proposal Bart which used neural machine translation architecture and was transformer-based. They considered Bart as generalizing Bert. It applied the bi-directional encoder, GPT, and more recent pre-training schemes. They evaluated several de-noising approaches finding the best performance. Bart is effective for fine-tuned text generation as well as for comprehension tasks [34].

Lei Sha, et. al. 2022 said that the core of original transformers was self-attention. It was a dot-product-based token-by-token correlation computation module. It built a connection between tokens. They considered the self-attention that focused on the current token as a disadvantage. They suggested a new novel architecture called Bet (bird-eye transformers) for providing transformers with high-level historical tokens. They encouraged the attention weights with some syntax guidance. They proposed two alternative architectures; a syntax-guided transformer architecture and a sytax-guidance-free transformer architecture [35].

Lei Liu, et. al. 2022 proposed a lightweight language model based on both; text augmentation technology and knowledge understanding mechanism. Their lightweight language model; EDA-BoB in which EDA stands for easy data augmentation, and BoB is stands for Bert over Bert model. Bert is the improved pre-trained language model. It provides the ability of text generation keeping in constraint the knowledge understanding mechanisms. Firstly, EDA boosted the annotated training sample data. Secondly, it is the role of a multi-Bert combination scheme to solve the two tasks; knowledge understanding, and text generation [8].

Xiaobo Liang, et. al. 2023 explored the potential of pre-trained masked language models (MLMs) alongside BART and GPTs, which have dominated open-ended long text generation (Open-LTG). However, the AR nature of these models reduces inference efficiency as the length of the generated text increases, limiting their application in Open-LTG. Preliminary findings indicate that pre-trained MLMs can only generate short texts and tend to collapse when modeling long texts. To enhance the long text generation capabilities of MLMs, the study introduced two simple yet effective strategies for iterative NAR models: dynamic sliding window attention (DSWA) and linear temperature decay (LTD) [36].

Yuhong Mo, et. al. 2024 developed a tool for detecting LLM AI text generation based on the Transformer model, aiming to improve the accuracy of AI text generation detection and provide a reference for subsequent research. Firstly, the text is Unicode normalized, converted to lowercase form, characters other than non-alphabetic characters and punctuation marks are removed by regular expressions, spaces are added around punctuation marks, first and last spaces are removed, consecutive ellipses are replaced with single spaces, and the text was connected using the specified delimiter. Next, non-alphabetic characters and extra whitespace characters were removed, multiple consecutive whitespace characters were replaced with a single space, and the text was converted to lowercase form again—the deep learning model combined layers such as LSTM [37].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1: Comparison between techniques** | |  |  |  |
| **Category** | **Description** | **Examples** | **Strengths** | **Weaknesses** |
| **Linguistic Grammar Techniques** | Based on predefined rules, conditions, and grammar for generating text. Focuses on rule-based, class-based, or statistical models. | Rule-based text generation, class-based systems, statistical models, predicate calculus | Explanatory, transparent decision-making | High human intervention, requires extensive manual work, knowledge collection, and repair. |
| **Machine and Deep Learning** | Utilizes neural networks and models such as RNN, CNN, and transformer models. Often leverages large datasets to achieve state-of-the-art text generation results. | Sequence-to-sequence models, GANs, Transformers, RNN, BERT, GPT | High performance, can handle complex text generation tasks, adaptive to varied contexts. | Poor explaining ability requires large training datasets and can be poorly behaved in certain scenarios. |
| **Lexically-Constrained Generation** | Generates text with specific constraints (e.g., keywords or contextual relevance). Commonly uses beam-search-based and stochastic searching methods. | Constrained Beam Search, Gradient-guided optimization | Flexibility, good for specific tasks that require constraints, such as keyword inclusion | This may limit creativity or diversity in the generated text. |
| **Hybrid Mechanisms** | Combines deep learning with knowledge understanding and syntactic guidance. Leverages pre-trained language models and fine-tuning for text generation tasks. | BERT, OpenAI GPT, Syntax-guided Transformers, BART, Bet, EDA-BoB | Effective for long text generation, integrates both knowledge and text augmentation strategies. | Requires complex architectures and computational power |

1. **Evaluation of Generated Texts**

As shown in this article, a variety of applications including the generation of text, as well as a variety of techniques, may be used for the generation. However, the measure to evaluate any technique is the generated text itself. Firstly, the proposed approach should be tested by applying it to known examples. Then, computing the similarity between the known text and the generated text [24, 38].

There are many methods applied for text similarity. Prakoso D.W., et, al. 2021 presented a comprehensive systematic review on text similarity. They showed the existing techniques and classified them. They found out points of strengths as well as weaknesses. They studied all possible effectors such as; language independent, domain-independent, corpus and training data, semantic knowledge, semantic meaning, and word order similarity [39].

Yuan W., et. al. 2021 thought that models trained to generate texts will be the best when the generated text is better. They considered that it is a challenge for text generators to evaluate whether the generated text is accurate, actually fluent, or effective. They proposed their metric Bart-Score which built on Bart [43] mentioned in a previous section of this article. Their code to calculate Bart-Score is available on the World Wide Web, as well as their interactive leaderboard for meta-evaluation. They succeeded in understanding the weaknesses, strengths, and complementarity of each metric [40].

Iqbal T. and Qureshi S. 2022 talked in their survey about the evaluation methods of text generation. They presented the following; [24].

* Rouge, the recall-oriented understudy for Gisting evaluation. The similarity between the reference summary and the system summary. They calculated the precision and the recall.
* Bleu, the bilingual evaluation under-study [41]. Bleu was used for machine-translated text, by calculating its similarity to the reference text via unigram and bi-gram. Bleu achieved more matches resulting in better machine translation.
* They presented many methods for evaluating generated text such as Banerjee and Lavie, Vedantam, et. al., 2015 [42], Anderson, et. al. 2016 [44], Semeniuta, et. al., 2018 [45] and Tevet, et. al. 2018 [46].

Although of all those methods, they considered the evaluation of generated text as an open research problem. Because no one of the mentioned methods was perfect and comprehensive.

**References**

1. Karanikolas N, Manga E, Samaridi N, Tousidou E, Vassilakopoulos M. Large language models versus natural language understanding and generation. InProceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics 2023 Nov 24 (pp. 278-290).
2. Becker J, Wahle JP, Gipp B, Ruas T. Text generation: A systematic literature review of tasks, evaluation, and challenges. arXiv preprint arXiv:2405.15604. 2024 May 24.
3. Abeer K. Al-Mashhsdany, and Safa A. Ahmed, “Textual Analysis Applications: Subject Review”, J. Of University Of Anbar For Pure Science, v.12, no.3, 2018.
4. Mou Lili and Vechtomova Olga, "Stylized Text Generation: Approaches and Applications", In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts", Association for Computational Linguistics pp. 19—22, 2020.
5. X. He and L. Deng, "Deep Learning for Image-to-Text Generation: A Technical Overview," IEEE Signal Processing Magazine, v. 34, no. 6, pp. 109-116, 2017.
6. Jacob Devlin, et al., 2019. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In NAACL-HLT.
7. [Qiuyun Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang%2C+Q), et al., “AI-Powered Text Generation for Harmonious Human-Machine Interaction: Current State and Future Directions”, [arXiv:1905.01984, v. 1](https://arxiv.org/abs/1905.01984v1)[cs.CL], 2019.
8. Lei Liu, et al., “Research and Implementation of Text Generation Based on Text Augmentation and Knowledge Understanding”, Hindawi, Computational Intelligence and Neuroscience, ID 2988639, 2022.
9. Sheng E, Chang KW, Natarajan P, Peng N. Societal biases in language generation: Progress and challenges. arXiv preprint arXiv:2105.04054. 2021 May 10.
10. Shi Zhan, et al., "Toward diverse text generation with inverse reinforcement learning." arXiv preprint arXiv:1804.11258, v3, 2018.
11. Laura Illia, Elanor Colleoni, and Stelios Zyglidopoulos, “Ethical implications of text generation in the age of artificial intelligence,” Business Ethics, the Environment & Responsibility, v.32, no.1, pp. 201-210, 2022.
12. Kulkarni, A., Shivananda, A., Kulkarni, A. (2022). Text Generation: Next Word Prediction. In: Natural Language Processing Projects . Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7386-9\_11
13. Chang Ernie, et al., "Neural Data-to-Text Generation with {LM}-based Text Augmentation", in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume", "Association for Computational Linguistics", 2021.
14. Hongyu Zang and Xiaojun Wan, “Towards Automatic Generation of Product Reviews from Aspect-Sentiment Scores”, Proceedings of The 10th International Natural Language Generation Conference, pages, Santiago de Compostela, Spain, September 4-7 2017. Association for Computational Linguistics, pp. 168–177, 2017.
15. Hanqing Zhang, et al., “A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models”, Association for Computing Machinery, v. 37, no. 4, pp. 111:1-111:38, 2022.
16. Dong C, Li Y, Gong H, Chen M, Li J, Shen Y, Yang M. A survey of natural language generation. ACM Computing Surveys. 2022 Dec 23;55(8):1-38.Wei Wei, Bei Zhou, and Georgios Leontidis, “A Hybrid Natural Language Generation System Integrating Rules and Deep Learning Algorithms”, arXiv:2006.09213v2 [cs.CL] 17, 2020.
17. Sha Lei, "Gradient-guided Unsupervised Lexically Constrained Text Generation", in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020
18. Abeer K.Al-Mashhsdany, Abdulwadood K.Al-Mashhadany, and Waleed K.Al-Mashhadany, “Root-Stem Approaches in General Analyzer System for Arabic Language (RSGAS)”, Journal of University of Anbar for Pure Science, v.10, no.3, pp. 41-49, 2016.
19. Gkatzia D., Lemon O., and Rieser V., 2017, “Data-to-Text Generation Improves Decision-Making Under Uncertainty”, IEEE Computational Intelligence Magazine, v. 12, no. 3, pp. 10-17.
20. Bauer, Armin, et al. “Rule-Based Approach to Text Generation in Natural Language - Automated Text Markup Language (ATML3).” Rules and Rule Markup Languages for the Semantic Web, 2015.
21. Emma Manning, “A Partially Rule-Based Approach to AMR Generation”, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, Minneapolis, Minnesota, June 3 - 5, 2019, Association for Computational Linguistics, pp. 61–70, 2019.
22. Shuohua Zhou, “Research on the Application of Deep Learning in Text Generation”, Journal of Physics: Conference Series 1693, 2020.
23. Touseef Iqbal, Shaima Qureshi, “The survey: Text generation models in deep learning”, Journal of King Saud University –Computer and Information Sciences, v.34 pp. 2515-2528, 2022
24. Wang Heng, Zengchang Qin, and Tao Wan. "Text generation based on generative adversarial nets with latent variables." In Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22. Springer International Publishing, pp. 92-103, 2018.
25. Qader, Raheel, François Portet, and Cyril Labbé. "Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models." arXiv preprint arXiv:1910.03484 (2019).
26. Bayer, Markus, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers." International Journal of Machine Learning and Cybernetics (2022): 135-150.
27. Ziang Xie, “Neural Text Generation: A Practical Guide”, [zxie@cs.stanford.edu](mailto:zxie@cs.stanford.edu), 2018 <http://cs.stanford.edu/~zxie/textgen.pdf>
28. Qin Lianhui, et. al., “COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics”, print arXiv:2202.11705, 2022.
29. Anderson Peter, et al., "Guided Open Vocabulary Image Captioning with Constrained Beam Search", in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, Association for Computational Linguistics, 2017
30. Qin Lianhui, et. al., “COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics”, print arXiv:2202.11705, 2022.
31. Junyi Li, et al., “Pre-trained Language Models for Text Generation: A Survey”, Association for Computing Machinery, v.1, no.1, 2022.
32. Alec Radford, et al., 2019. “Language models are unsupervised multitask learners”, Open AI blog, 2019.
33. Mike Lewis, et al., “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. arXiv preprint arXiv:1910.13461, 2019
34. Sha Lei, et al., "Bird-Eye Transformers for Text Generation Models", arXiv preprint arXiv:2210.03985, v.1, 2022.
35. Xiaobo Liang, Zecheng Tang, Juntao Li, and Min Zhang. 2023. “ [Open-ended Long Text Generation via Masked Language Modeling](https://aclanthology.org/2023.acl-long.13)”, In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 223–241, Toronto, Canada. Association for Computational Linguistics.
36. Y Mo, [H Qin](https://scholar.google.com/citations?user=3ITPmYUAAAAJ&hl=ar&oi=sra), Y Dong, [Z Zhu](https://scholar.google.com/citations?user=WS9nHAQAAAAJ&hl=ar&oi=sra), Z Li‏.“ Large Language Model (LLM) AI text generation detection based on transformer deep learning algorithm”, arXiv preprint arXiv:2405.06652, 2024‏•arxiv.org‏
37. Weizhe Yuan, Graham Neubig, and Pengfei Liu, “BARTScore: Evaluating Generated Text as Text Generation”, 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.,arXiv:2106.11520v2 [cs.CL], 2021.
38. Prakoso, D.W., Abdi, A. & Amrit, C., “Short text similarity measurement methods: a review”, Soft Computing, v.25, pp. 4699–4723, 2021.
39. Weizhe Yuan, Graham Neubig, and Pengfei Liu, “BARTScore: Evaluating Generated Text as Text Generation”, 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.,arXiv:2106.11520v2 [cs.CL], 2021.
40. Papineni K., et al., “Bleu: a method for automatic evaluation of machine translation”. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 311–318, 2002.
41. Vedantam R., Lawrence C. and Parikh D., 2015. “Cider: Consensus-based image description evaluation”, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4566–4575.
42. Mike Lewis, et al., “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. arXiv preprint arXiv:1910.13461, 2019
43. Anderson P., et al., “Spice: Semantic propositional image caption evaluation”, In European Conference on Computer Vision. Springer, pp. 382–398, 2016.
44. Semeniuta S., Severyn A., and Gelly S., “On the accurate evaluation of gans for language generation”, arXiv preprint arXiv:1806.04936, 2018.
45. Tevet G., et al., “Evaluating text gans as language models”, arXiv preprint arXiv:1810.12686, 2018.