International Conference on Machine Learning and Data Engineering

**Multi-Head Attention Transformer for Text2Text Translation**

Shivakumar Ba, Sandeep Ub, Varshini Vc, Kumaran Pd, \*

a,b,c,dNational Institute of Technology Puducherry, Thiruvettakudy, Puducherry 609609, India

Abstract

In this paper, we focus on the application of transformer models in English-to-Tamil translation and mainly their huge contribution to the task of Text2Text translation. We underline the strengths of the transformer architecture while at the same time emphasizing how much self-attention mechanisms allow for very effective treatments of long-range dependencies and contextual nuances in translations that are very relevant, especially in morphologically rich languages like Tamil. We will use BLEU as a measure to compare the translations from these models. BLEU is one of the most used metrics, measuring how close machine-generated outputs are to human reference translations. Comparative analyses against traditional machine translation methods show transformer models surpassing previous techniques in producing fluent and contextually sensitive translations. Finally, this research underlines the potential impact of state-of-the-art deep learning methods on language translation tasks and points out future opportunities for the improvement of MT performance within low-resource languages.

© 2025 The Authors. Published by ELSEVIER B.V.

This is an open access article under the CC BY-NC-ND license ([https://creativecommons.org/licenses/by-nc-nd/4.0](https://nam11.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcreativecommons.org%2Flicenses%2Fby-nc-nd%2F4.0&data=05%7C01%7CPROCS%40elsevier.com%7C62360667222a4d5f398b08db29f20f6b%7C9274ee3f94254109a27f9fb15c10675d%7C0%7C0%7C638149891839468101%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=YiZJOJH5ZqyE58SojxNCuUtjAhnNezZeuJK9WmAdNYY%3D&reserved=0))

Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering

*Keywords:* Text2Text Translations, Transformer, Neural Networks, Low-resource language, Multihead Attention

**1. Introduction**

Language is a tool that is used to communicate information across different cultures found all over the globe. Nonetheless, language barriers often hinder effective interaction between diverse cultural groups. Machine translation has become an essential technology to surpass this challenge and translate text or speech from one language to another. The first attempts in machine translation were dictionary-based, where the system utilized language dictionaries to search for words in the target language. This paradigm was prevalent from the late 1940s until the mid-1960s. As technology evolved, new approaches led to the creation of Rule-Based Machine Translation. The RBMT includes maximum information about linguistic aspects, and detailed knowledge concerning morphology, syntax, and semantic levels in both source and target languages come into play, and this would yield more accurate and contextually relevant translations.

Statistical Machine Translation The next step in the process, powers the more familiar systems, including Google Translate™, Microsoft Translator, and Asia Online™. Not like the first MT systems that depended on pre-made dictionaries and grammar, SMT uses statistical models to translate text based on the patterns learned from large bilingual datasets [2].

.

Thanks to the low computing power and bad quality of the first models, research in neural network-based translation was neglected for a long time. With the rise of Deep Learning in the 2010s, many NLP tasks started to improve significantly. Accordingly, the application of deep neural networks in machine translation has received high attention recently. Among the very first to introduce a working approach for NMT were Kalchbrenner and Blunsom.

A very important step in this evolution was the appearance of a model called Transformer, proposed by Vaswani et al., who implemented self-attention mechanisms to increase translation accuracy and efficiency. Their work showed that transformers can be more accurate and efficient than earlier models since they process the sentence as one piece, hence capturing long-range dependencies better.

In this paper, we outline the experimental setup and results obtained from training the Transformer model on the English-Tamil subset of the Samanantar dataset, which is the largest publicly available parallel corpus for Indic languages. This subset contains 1.5 million parallel records, introduced in the paper of Kunchukuttan et al. [6]. The rest of the paper is as follows: Section 2 covers the various related works that inspire our study. Section 3 describes the model and methodology, Section 4 provides the comparative analysis of the results in different training configurations, and Section 5 concludes the paper.

**2. Architecture**

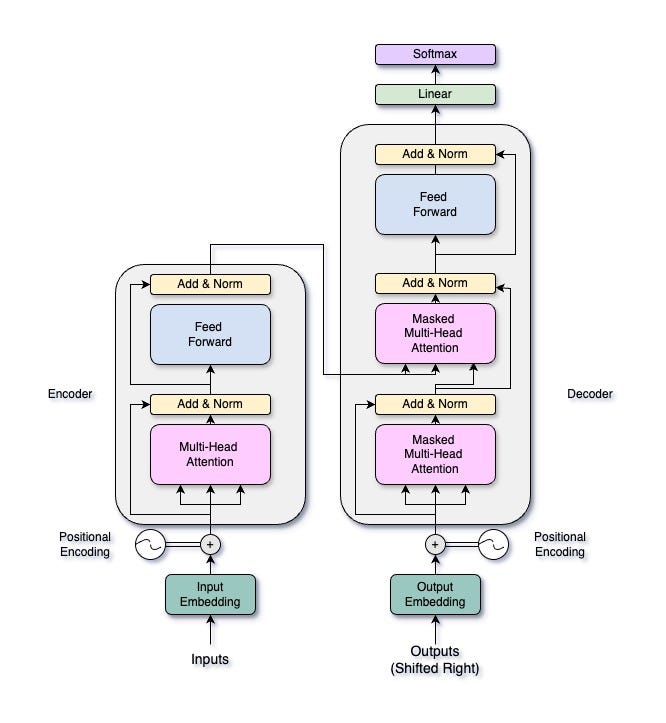


Fig 2. Attention mechanism [5]

A diagram of a software algorithm

Description automatically generated

Fig 1. Model architecture

**3 Literature Review**

"Attention Is All You Need" delineates the transformer architecture, which relies exclusively on self-attention mechanisms, discarding recurrence for a parallelizable model. Concretely, their architecture sticks to the encoder-decoder structure, where the encoder works on the input sequence to create a set of attention-based representations, while on the other hand, it is the decoder that generates the output sequence. It has multiple attention heads in parallel that learn different representation subspaces; this also includes positional encoding and layer normalization, among others, which help the model catch the context-based relationships in the data [5].

On this ground, BERT proposes a method of pre-training language representations that can be fine-tuned for particular tasks. Masked language modeling and next-sentence prediction are used to enable the transformer architecture under BERT to capture bidirectional context. In this way, it also shows both the strength of transfer learning and fine-tuning in the area of natural language processing with huge gains on most tasks[7].Another key technique that could further stabilize training in a deep neural network is Layer normalization. While Batch Normalization normalizes across the batch dimension, Layer Normalization normalizes across features for each individual training example. This prevents internal covariate shift and allows the model to learn more efficiently on top, especially in recurrent architectures.

For transformers, it is quite necessary to have position encoding to be able to capture the sequential nature of the input data because of the lack of recurrence in the architecture. Several studies have been conducted on different techniques for position encoding. Some evolved from a direct addition of sinusoidal or learnable positional vectors to the input embeddings. In this way, it would enable the model to make out the order of tokens in a sequence, enhancing the representational power of the transformer. You may use a specific reference to add weight to the discussion of this point[5].

In the domain of machine translation, "Neural Machine Translation by Jointly Learning to Align and Translate" describes an attention mechanism that explicitly allows the model to focus on specific parts of the input sequence dynamically while decoding. The authors show that this helps in improving translation quality by aligning the encoder hidden states to the decoder output, resulting in more accurate and context-aware translations[9].

Attention mechanisms, first developed for NLP, are now one of the central components in deep learning models that enable model focus on features considered relevant from complicated datasets. Attentional mechanisms accomplish -this by enhancing a model's ability to learn long-range dependencies and contextual relationships through dynamic weighting of inputs. Applications are not limited to NLP; examples extend into areas such as computer vision and speech recognition, even into modeling complex systems. With the development of research, attention mechanisms have been proven versatile and have given new insights and methods to approach complex problems in many domains[10].

In 2002, Papineni et al. put forward the BLEU metric, which is a pioneering method for the automated evaluation of machine translation [11]. BLEU scores the degree of similarity between a machine-generated translation and the reference translation by computing n-gram precision combined with a brevity penalty to punish overly short translations. It became very popular because it provided fast and reliable assessment of the quality of translations, hence the possibility of comparing and developing different machine translation models. BLEU's introduction has had a lasting impact on the field by becoming the standard evaluation metric in research and industry alike.

**4. Methodology**

Our methodology leverages the Transformer architecture, specifically its self-attention mechanisms, to improve the accuracy and efficiency of English to Tamil translation. By focusing on careful preprocessing, model training, and evaluation, we demonstrate the efficacy of advanced deep learning techniques in language translation, particularly for low-resource languages.

*4.1 Input Tokenization and Embedding*

*4.1.1 Tokenization & Embedding*

The first step in our process is **tokenization**, which involves breaking down the input sentences into smaller units known as tokens. These tokens can represent words, sub-words, or even characters, depending on the chosen granularity. Tokenization is crucial as it converts natural language text into a structured format that the model can process. After tokenization, each token is mapped to a unique high-dimensional vector through **embedding**. This transformation from token indices to dense vectors captures the semantic meaning of each token in a continuous vector space, providing a rich representation that facilitates the model's understanding of the text

*4.1.2 Position Encoding*

To incorporate positional information, positional encoding is added to the token embeddings sequence[5]. The position encoding P(i,d) for the **i-**th position and **d**-th dimension is calculated as:

(1)

(2)

where **k** is the dimension index.

The final input representation after adding positional encoding to the embedding is:

(3)

*4.2 Encoder*

The encoder processes the input sequence to generate a set of hidden representations through self-attention mechanisms. Each layer in the encoder consists of multi-head self-attention followed by a position-wise fully connected feed-forward network[5].

*4.2.1 Feed Forward Layer*

The token embeddings with positional encoding are passed through a feed-forward neural network within the encoder layer. The feed-forward layer transforms the embeddings by applying learned weights and biases to produce a higher-dimensional representation, enabling the model to capture complex relationships within the sequence as :

**FFN(Zi​)=max (0, Zi-W1​+b1​)⋅W2​+b2** (4)

where **W**1​, **W**2​, **b**1​, and **b2** ​are learned weights and biases.

*4.2.2 Projection into Q, K, and V Matrices*

The output of the feed-forward layer is projected into three matrices: Query (Q), Key (K), and Value (V) as:

**Q=Z ⋅ Wq, K = Z ⋅ Wk, V = Z ⋅ WV​**  (5)

where **W**q​, **Wk**​, and **Wv**​ are the weight matrices for queries, keys, and values, respectively.

*4.2.4 Scaled Dot-Product Attention*

The self-attention mechanism[5] calculates attention scores by computing the dot product of the query (q) and the transpose of the key (k) matrices, divided by square root of dimensionality of key vectors(dk) as :

(6)

Where the attention output is:

**Attention (Q, K, V) =A⋅V**  (7)

where dk​ is the dimensionality of the key vectors. This process allows the model to weigh the importance of each token relative to the others, enabling it to focus on the most relevant parts of the input sequence.

*4.2.5 Concatenation and Linear Transformation*

The outputs from the multiple attention heads are concatenated to form a single matrix. This concatenated output is then passed through a linear layer, which combines the information from all attention heads into a unified representation as :

**H= [head1; head2​…;headn​]⋅W0** (8)

where **W**0 is the output weight matrix, and **h** is the number of heads.

*4.2.6 Add & Normalize*

A residual connection is applied by adding the original input to the output of the self-attention mechanism as :

**Z′=LayerNorm(Z+H)** (9)

where **Z** is the original input, and **H** is the output from the attention mechanism.

*4.3 Decoder*

The decoder uses the encoder's hidden representations along with its own self-attention mechanisms to generate the target sequence. It incorporates an additional layer of attention over the encoder's output to focus on relevant parts of the input sentence during translation.

*4.3.1 Masked Multi-Head Self-Attention*

In the decoder, a multi-head self-attention mechanism is employed with masking **M** to prevent the model from attending to future tokens during the generation of the current token. adhering to this constraint, enabling left-to-right sequence generation as :

(10)

*4.3.2 Encoder-Decoder Attention*

The decoder layer incorporates encoder-decoder attention, where the encoder's output serves as the keys and values, while the decoder's output serves as the queries as :

(11)

where **Q**dec ​ is the query from the decoder, and **K**enc and **V**enc​ are the key and value from the encoder.

*4.3.3 Add & Normalize*

Like the encoder, residual connections and layer normalization are applied after each attention mechanism.

**5. Results and performance analysis**

*5.1 Experimental Setup*

This proposed transformer has been applied to the dataset which contains English to Tamil sentence pairs collected from the Samanantar Indic Language dataset[6] . The size of the collected dataset exceeds more than 5 million data pairs. The used dataset has been taken from multiple sources like newspapers (The times of india, Good returns etc..), education domain (NPTEL, Khan Academy, etc..). For extraction of data from mainstream webistes they’ve used Beautiful Soap or Selenium. For Youtube videos and other video sources they’ve collected subtitles available on **March 7th 2021**[6]

For the construction of the transformer, we’ve used the pytorch framework in python and implemented the transformer using the Neutral Network module (nn.Module). Pytorch is an Machine Learning library based on the torch library. It was originally developed by Meta AI now it is a part of the Linux Foundation umbrella. Since, we own a NVIDIA RTX 3060 GPU(Graphical Processing Unit) we ran the model on the CUDA(Compute Unified Device Architecture) mode.

The given dataset has been pre-processed into the following state:

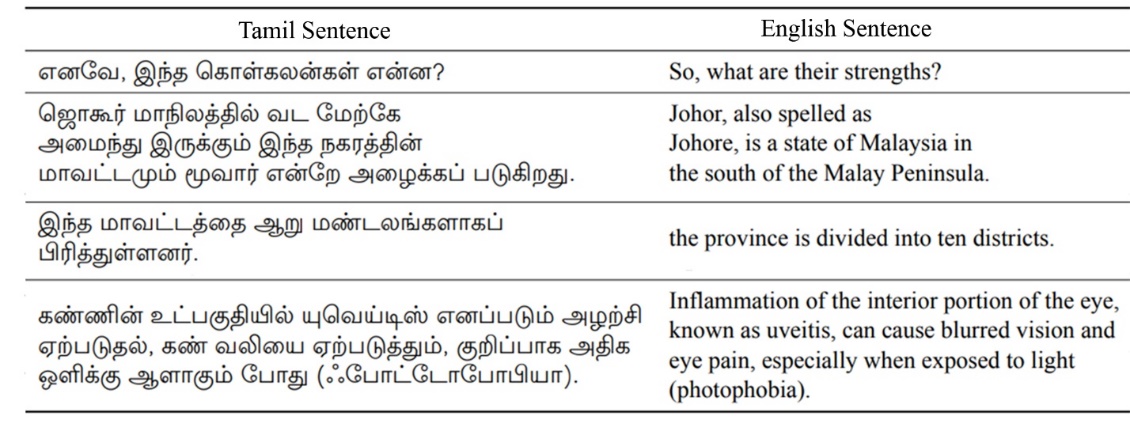


Table 1 Data format for the experiment[6]

For the mathematical part of the program, we’ve used the Numpy framework available in python. Numpy is a library that adds support to python for large, multi-dimensional arrays and matrices. Using numoy we can calculate the matrice multiplication values faster than ever(eq; qkv calculations).

*5.2 Performance Metrics*

For translation, the accuracy of the system is used to justify the appropriateness of the transformer for the specified task, BLEU Score[11] is used to evaluvate the performance.

*5.2.1 BLEU Score*

BLEU score is an evaluvation metric for Machine Translation tasks. It is calculated by comparing the n-grams of machine-translated sentences and n-grams of human-translated sentences. It can be noted that BLEU Score decreases as the sentence length increases[11]. BLEU score has a range of [0,1].

BLEU Score can be calculated by:

(12)

BP 🡪 Brevity Penalty

wn 🡪 Weight for n-gram precision of order n

pn 🡪 modified n-gram precision score of order n

N 🡪 Maximum n-gram order to consider

Brevity Penalty(BP)

(13)

r 🡪 length of the candidate translation

c 🡪 average length of the reference translations

Modified n-gram precision(pi)

(14)

Count Clip 🡪 Function that clips number of matched n-grams divided by total number of n-grams

matchesi 🡪 number of n-grams of order I that match exactly between the candidate translation and reference translations

max-ref-count 🡪 maximum number of occurrences of the specific n-gram of order i found in any single reference translation candidate-n-gramsi 🡪 total number of n-grams of order i present in the candidate translation

*5.3 Experimental results*

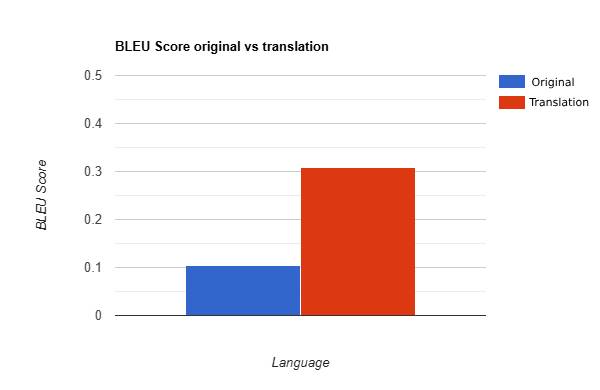


Fig 3. BLEU Score original vs translation

From the above graph we can infer that before the translation the BLEU score between the English sentence and the expected translations are less than 0.1. This is due to the words i.e., the tokens in every sentence pair having some tokens of the same length. Since, BLEU score takes into consideration both the token length and the token similarity to consideration.

We can also note that the translated language has a BLEU Score of 0.30857. Generally, any BLEU score above 2.5 is considered good for a translator. However, some exceptions can be found here since, the words are sometimes jumbled causing reductions in the score. For example; The sentence **“What is your name?”** has a Tamil translation of **“உங்கள் பெயர் என்ன?”** but our transformer when asked to translate the given sentence gives out the sentence as **“உங்கள் என்ன பெயர்?”**  which has the last two words of the sentence exchanged. We can reduce mishaps like these by training the model for more epochs or giving it an even wider dataset. For accomplishing both these tasks we would need more computing power and time.

We can notice that the BLEU Score increases as the no.of n-grams of the original text increases. Since, it is easy to translate one word directly the translator performs better when given a word but, when given a sentence containing multiple words it tries to understand the context of the sentence and tries to give out an output accordingly. Hence, each words play a role in the sentence trying to find which word has more preference and which words don’t can be improved by training the model for longer times. For, example when trying to translate the sentence “**Thanks”** our model gives out the correct answer as **“நன்றி”** but when asked to translate the sentence “**Thank you”** It gives the output as “**நன்றி நன்றி நன்றி**” even though the expected output is just **“நன்றி”**. Since both “Thanks” and “Thank you” have the same translation and since, our transformer stems thanks and thank as thank the transformer gives out the same output as நன்றி.

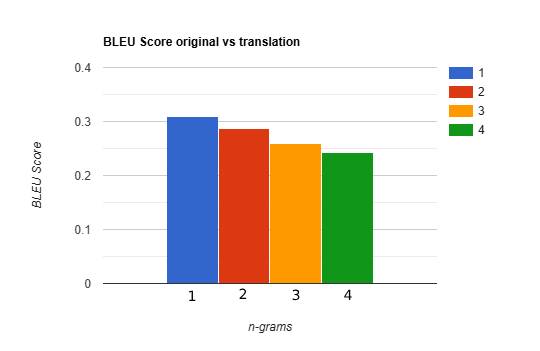


Fig 4. BLEU Score based on n-grams

For calculation of the BLEU score we’ve used NLTK’s BLEU score framework which calculates the BLEU score when given the results and the expected results. For evaluvation of the model we used 10 percent data split of the 90:10 training split of the dataset we’ve taken.

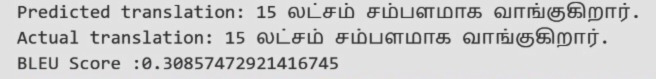


Fig 5. BLEU score analysis with the test dataset

The model produced translations with a BLEU score of 0.30857, which aligns fairly well with human reference translations. That score very clearly reflected basic handling of the core aspects of the Tamil language, its grammar, and basic contextual nuances—one of the toughest tasks in a language as morphologically complex as Tamil.

The performance is commendable, especially when translation is done between a rich resource language like English and a poor resource language like Tamil. On the contrary, the BLEU score will be high if the model works fine most of the time but leaves some scope for enhancement with respect to complex sentences and idiomatic expressions.The sample translation is as shown below, which the model can generate so that in-context and coherent translations are properly placed. Outputs such as the ones above emphasize the promise of transformer-based models in the field of neural machine translation, specifically for low-resource languages.

**6. Conclusion**

In this work, we have developed a neural machine translation model that translates text from English into Tamil with the Transformer architecture. This model performed at a BLEU score of 0.30857, handling very complex linguistic patterns in Tamil—a language very rich in morphology. It has been trained on a dataset of 100,000 phrase pairs for 30 epochs. It is especially capable of comprehending long-range relationships and fine-grained context, very important for high-quality translations, through the use of multi-head self-attention mechanisms. Though having huge potential, this model does not work well with idiomatic expressions and intricate grammatical structures. It means it needs more fine-tuning to achieve better results on other more complex models of phrases. In the future, this could be done using even larger datasets, addition of more language-specific characteristics, and methods such as transfer learning that can be implemented to improve the model's accuracy. The results show that the use of transformer models is very useful in neural machine translation for low-resource languages. This would then suggest a few ways whereby the quality of translations can be improved toward real-world applications.

**References**

[1]Tripathi, Sneha & Sarkhel, Juran. Approaches to machine translation. Annals of Library and Information Studies. 57. 388-393. (2010)

[2]Kenny, Dorothy & Doherty, Stephen. (2014). Statistical machine translation in the translation curriculum: Overcoming obstacles and empowering translators. The Interpreter and Translator Trainer. 8. 276-294. 10.1080/1750399X.2014.936112.

[3]Badawi, S. (2023). A Transformer-Based Neural Network Machine Translation Model for the Kurdish Sorani Dialect. *UHD Journal of Science and Technology*, *7*(1), 15–21. <https://doi.org/10.21928/uhdjst.v7n1y2023.pp15-21>

[4] Nal Kalchbrenner and Phil Blunsom. 2013. .Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics. <https://aclanthology.org/D13-1176.pdf>

[5] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017). <https://arxiv.org/abs/1706.03762v7>

[6] Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages

[Gowtham Ramesh](https://arxiv.org/search/cs?searchtype=author&query=Ramesh,+G), [Sumanth Doddapaneni](https://arxiv.org/search/cs?searchtype=author&query=Doddapaneni,+S), [Aravinth Bheemaraj](https://arxiv.org/search/cs?searchtype=author&query=Bheemaraj,+A), [Mayank Jobanputra](https://arxiv.org/search/cs?searchtype=author&query=Jobanputra,+M), [Raghavan AK](https://arxiv.org/search/cs?searchtype=author&query=AK,+R), [Ajitesh Sharma](https://arxiv.org/search/cs?searchtype=author&query=Sharma,+A), [Sujit Sahoo](https://arxiv.org/search/cs?searchtype=author&query=Sahoo,+S), [Harshita Diddee](https://arxiv.org/search/cs?searchtype=author&query=Diddee,+H). <https://arxiv.org/abs/2104.05596>

[7] Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *North American Chapter of the Association for Computational Linguistics* (2019). <https://arxiv.org/abs/1810.04805>

[8] Ba, Jimmy et al. “Layer Normalization.” *ArXiv* abs/1607.06450 (2016): n. pag. <https://arxiv.org/abs/1607.06450>

[9] Bahdanau, Dzmitry et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *CoRR* abs/1409.0473 (2014): n. pag. <https://arxiv.org/abs/1409.0473>

[10] Hernández, Adrián & Amigó, José. (2021). Attention Mechanisms and Their Applications to Complex Systems. Entropy. 23. 283. 10.3390/e23030283. <http://dx.doi.org/10.3390/e23030283>

[11] Bleu: a Method for Automatic Evaluation of Machine Translation] (<https://aclanthology.org/P02-1040>) (Papineni et al., ACL 2002)

[12] Large Language Models for Software Engineering: Survey and Open Problems [Angela Fan](https://arxiv.org/search/cs?searchtype=author&query=Fan,+A), [Beliz Gokkaya](https://arxiv.org/search/cs?searchtype=author&query=Gokkaya,+B), [Mark Harman](https://arxiv.org/search/cs?searchtype=author&query=Harman,+M), [Mitya Lyubarskiy](https://arxiv.org/search/cs?searchtype=author&query=Lyubarskiy,+M), [Shubho Sengupta](https://arxiv.org/search/cs?searchtype=author&query=Sengupta,+S), [Shin Yoo](https://arxiv.org/search/cs?searchtype=author&query=Yoo,+S), [Jie M. Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang,+J+M) (<https://arxiv.org/pdf/2310.03533>)

[13] DALE: Generative Data Augmentation for Low-Resource Legal NLP

[Sreyan Ghosh](https://arxiv.org/search/cs?searchtype=author&query=Ghosh,+S), [Chandra Kiran Evuru](https://arxiv.org/search/cs?searchtype=author&query=Evuru,+C+K), [Sonal Kumar](https://arxiv.org/search/cs?searchtype=author&query=Kumar,+S), [S Ramaneswaran](https://arxiv.org/search/cs?searchtype=author&query=Ramaneswaran,+S), [S Sakshi](https://arxiv.org/search/cs?searchtype=author&query=Sakshi,+S), [Utkarsh Tyagi](https://arxiv.org/search/cs?searchtype=author&query=Tyagi,+U), [Dinesh Manocha](https://arxiv.org/search/cs?searchtype=author&query=Manocha,+D)(<https://arxiv.org/pdf/2310.15799>)

[14] Self-Attention Generative Adversarial Networks [Han Zhang](https://arxiv.org/search/stat?searchtype=author&query=Zhang,+H), [Ian Goodfellow](https://arxiv.org/search/stat?searchtype=author&query=Goodfellow,+I), [Dimitris Metaxas](https://arxiv.org/search/stat?searchtype=author&query=Metaxas,+D), [Augustus Odena](https://arxiv.org/search/stat?searchtype=author&query=Odena,+A) (<https://arxiv.org/pdf/1805.08318>)

[15] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota and A. Soomro, "A Systematic Literature Review on Text Generation Using Deep Neural Network Models," in IEEE Access, vol. 10, pp. 53490-53503, 2022, doi: 10.1109/ACCESS.2022.3174108.

[16] Pretrained Language Models for Text Generation: A Survey [Junyi Li](https://arxiv.org/search/cs?searchtype=author&query=Li,+J), [Tianyi Tang](https://arxiv.org/search/cs?searchtype=author&query=Tang,+T), [Wayne Xin Zhao](https://arxiv.org/search/cs?searchtype=author&query=Zhao,+W+X), [Ji-Rong Wen](https://arxiv.org/search/cs?searchtype=author&query=Wen,+J) (<https://arxiv.org/pdf/2105.10311>)

[17] Modern Methods for Text Generation [Dimas Munoz Montesinos](https://arxiv.org/search/cs?searchtype=author&query=Montesinos,+D+M) (<https://arxiv.org/pdf/2009.04968>)

[18] A Survey of GPT-3 Family Large Language Models Including ChatGPT and GPT-4 [Katikapalli Subramanyam Kalyan](https://arxiv.org/search/cs?searchtype=author&query=Kalyan,+K+S)(<https://arxiv.org/pdf/2310.12321>)