iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61633

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Understanding and Enhancing XLNet: A Comprehensive Exploration of Permutation Language Modeling

Dr. Pankaj Malik[1], Vidhi Gupta[2], Vanshika Vyas[3], Rahul Baid[4], Parth Kala
[1]*Asst. Prof.* [2, 3, 4, 5]*Student, Medi-Caps University, Indore*

Abstract: ***XLNet, a recent breakthrough in natural language processing, has garnered significant attention for its exceptional performance across various NLP tasks. At the core of XLNet lies Permutation Language Modeling (PLM), a novel approach that combines the strengths of autoencoding and autoregressive methods. This paper presents a comprehensive exploration of XLNet and its underlying PLM mechanism. We delve into the theoretical foundations of PLM, elucidate the XLNet architecture, and analyze its training procedure. Furthermore, we investigate strategies for enhancing XLNet's performance and efficiency, including parameter tuning, knowledge distillation, and domain adaptation. Experimental results on benchmark datasets validate the effectiveness of our proposed enhancements and provide insights into the future directions of XLNet-based research.***
Keywords: XLNet, Permutation Language Modeling, Autoencoding, Autoregressive, Natural Language Processing, Language Model, Transfer Learning

I. INTRODUCTION

Natural Language Processing (NLP) has witnessed remarkable advancements in recent years, largely driven by the development of powerful deep learning models. Among these models, XLNet stands out as a state-of-the-art architecture that has demonstrated exceptional performance across a wide range of NLP tasks. Central to XLNet's success is its innovative approach known as Permutation Language Modeling (PLM), which offers a unique blend of autoencoding and autoregressive methods.

Traditional language models, such as those based on autoregressive approaches like GPT (Generative Pre-trained Transformer), rely on left-to-right or right-to-left sequential modeling of text, limiting their ability to capture bidirectional context effectively. In contrast, XLNet's PLM leverages permutations of input sequences to model bidirectional context in a more flexible and comprehensive manner. By considering all possible permutations of the input tokens, XLNet achieves superior contextual understanding and generalization capabilities.

The introduction of XLNet and its PLM mechanism has sparked significant interest and research in the NLP community. Researchers and practitioners are keen to understand the underlying principles of XLNet, explore its applications across various NLP tasks, and devise strategies to further enhance its performance and efficiency. This paper aims to provide a comprehensive exploration of XLNet and its PLM mechanism, shedding light on its theoretical foundations, architecture, training procedure, and practical implications.

In this paper, we begin by elucidating the theoretical foundations of Permutation Language Modeling, comparing it to traditional autoencoding and autoregressive approaches. We then delve into the architecture of XLNet, discussing its key components and the training procedure used to optimize PLM objectives. Subsequently, we investigate strategies for enhancing XLNet's performance and efficiency, including parameter tuning, knowledge distillation, and domain adaptation.

Through extensive experimentation on benchmark datasets, we validate the effectiveness of our proposed enhancements and provide insights into the strengths and limitations of XLNet in real-world applications. Our findings contribute to a deeper understanding of XLNet and its potential implications for the future of NLP research and applications.

Overall, this paper serves as a comprehensive guide to XLNet and Permutation Language Modeling, offering valuable insights into one of the most promising advancements in modern natural language processing.

II. THEORETICAL FOUNDATIONS OF PERMUTATION LANGUAGE MODELING

Permutation Language Modeling (PLM) is a novel approach to language modeling that forms the theoretical foundation of XLNet, a state-of-the-art natural language processing model.

©IJRASET: All Rights are Reserved

PLM combines elements of both autoencoding and autoregressive methods to achieve bidirectional context modeling in a flexible and comprehensive manner. In this section, we delve into the theoretical underpinnings of PLM, elucidating its key concepts and principles.

A. Autoencoding and Autoregressive Models

1) Autoencoding models, such as BERT (Bidirectional Encoder Representations from Transformers), learn contextualized representations by reconstructing input sequences from corrupted versions of themselves. They encode bidirectional context by masking input tokens and predicting missing tokens.
2) Autoregressive models, exemplified by models like GPT (Generative Pre-trained Transformer), generate text sequentially by predicting the next token based on preceding tokens. While autoregressive models capture left-to-right or right-to-left context, they lack bidirectional understanding.

B. Permutation-based Modeling

1) PLM introduces a novel permutation-based approach to language modeling, which overcomes the limitations of autoencoding and autoregressive methods. Instead of relying solely on sequential or masked input, PLM considers all possible permutations of the input tokens.
2) By sampling permutations of input sequences, PLM enables bidirectional context modeling without the need for sequential processing. Each permutation provides a unique view of the input sequence, allowing the model to capture diverse contextual information.

C. Bidirectional Context Modeling

1) Unlike autoregressive models, which are inherently unidirectional, and autoencoding models, which rely on masked input, PLM achieves bidirectional context modeling through permutation sampling. By considering all possible permutations, PLM captures bidirectional dependencies within the input sequence.
2) Bidirectional context modeling is essential for tasks requiring a deep understanding of context, such as language understanding, question answering, and sentiment analysis. PLM's ability to capture bidirectional context contributes to its effectiveness in various NLP tasks.

D. Objective Function and Training Procedure

1) During training, PLM aims to maximize the likelihood of generating the original input sequence from its permutations. The objective function encourages the model to assign high probabilities to permutations that preserve the input sequence's semantics.
2) PLM training involves sampling permutations of input sequences, passing them through the model, and optimizing the likelihood of generating the original input sequence. This process enables the model to learn robust contextual representations that capture bidirectional dependencies.

E. Flexibility and Generalization

1) PLM's permutation-based approach offers flexibility and generalization capabilities beyond traditional autoregressive and autoencoding methods. By considering diverse permutations, PLM learns representations that are invariant to permutation order and generalize well across various NLP tasks and domains.
2) The flexibility of PLM allows it to capture complex linguistic structures and dependencies, making it a versatile and powerful language modeling paradigm.

III. XLNET ARCHITECTURE AND TRAINING PROCEDURE

XLNet, a state-of-the-art natural language processing model, is built upon the Transformer architecture and Permutation Language Modeling (PLM) approach. In this section, we provide an overview of the XLNet architecture and detail its training procedure, which leverages PLM to learn contextualized representations from input sequences.

A. Transformer Architecture

1) XLNet employs the Transformer architecture, which consists of a stack of transformer blocks. Each transformer block comprises self-attention layers and feedforward neural networks.

2) Self-attention mechanisms allow the model to capture contextual relationships between tokens in the input sequence, while feedforward neural networks process and transform the token representations.
3) The transformer architecture facilitates parallel processing and efficient modeling of long-range dependencies, making it well-suited for language modeling tasks.

B. Permutation Language Modeling (PLM)

1) XLNet's key innovation lies in its adoption of Permutation Language Modeling (PLM), a novel approach that combines autoencoding and autoregressive methods.
2) During training, XLNet samples permutations of input sequences and aims to maximize the likelihood of generating the original sequence from its permutations.
3) By considering all possible permutations, XLNet captures bidirectional context in a flexible and comprehensive manner, leading to more robust and generalizable representations.

C. Training Procedure

1) XLNet training begins with initializing the model parameters randomly or using pretraining from a related task, such as masked language modeling (MLM) or autoregressive language modeling.
2) Input sequences are tokenized and segmented into fixed-length sequences or processed as streams of tokens. XLNet samples permutations of input sequences and feeds them into the model.
3) The model computes the likelihood of generating the original sequence from its permutations using the PLM objective function. Training involves optimizing this likelihood using gradient-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam.
4) During optimization, XLNet updates the parameters of the model to minimize the discrepancy between the predicted likelihood and the ground truth likelihood of the original sequence.

D. Fine-tuning and Transfer Learning:

1) After pretraining on a large corpus of text data, XLNet can be fine-tuned on downstream tasks using supervised learning techniques.
2) Fine-tuning involves adapting the pretrained XLNet model to specific tasks by updating its parameters on task-specific datasets. This process enables the model to learn task-specific patterns and representations.
3) Transfer learning from pretrained XLNet models has been shown to improve performance on various NLP tasks, including text classification, question answering, and natural language understanding.

E. Model Evaluation

1) XLNet's performance is evaluated on downstream tasks using task-specific evaluation metrics, such as accuracy, F1 score, or perplexity.
2) Evaluation involves assessing the model's ability to generate coherent and contextually relevant responses, make accurate predictions, or classify text into predefined categories.

IV. APPLICATIONS OF XLNET IN DOWNSTREAM TASKS

XLNet, with its innovative Permutation Language Modeling (PLM) approach and powerful Transformer architecture, has demonstrated remarkable performance across various downstream natural language processing (NLP) tasks. In this section, we explore the applications of XLNet in a range of tasks and highlight its effectiveness in each domain.

A. Text Classification

1) XLNet has been successfully applied to text classification tasks, including sentiment analysis, topic classification, and spam detection.
2) By fine-tuning pretrained XLNet models on labeled text data, researchers have achieved state-of-the-art performance in classifying text into predefined categories.

B. Question Answering

1) XLNet's bidirectional context modeling capabilities make it well-suited for question answering tasks, where understanding context is crucial for providing accurate answers.
2) Pretrained XLNet models have been fine-tuned on question answering datasets like SQuAD (Stanford Question Answering Dataset), achieving competitive performance in extracting answers from passages of text.

C. Named Entity Recognition (NER)

1) XLNet's contextualized representations enable it to capture nuanced information about named entities in text, making it effective for NER tasks.
2) Researchers have fine-tuned XLNet on NER datasets to extract entities such as person names, locations, organizations, and dates with high accuracy.

D. Machine Translation

1) XLNet's bidirectional context modeling and transfer learning capabilities make it a promising candidate for machine translation tasks.
2) Researchers have explored fine-tuning XLNet on multilingual translation datasets, achieving competitive performance in translating text between different languages.

E. Text Generation

1) XLNet's autoregressive capabilities make it suitable for text generation tasks, including language modeling, story generation, and dialogue generation.
2) Researchers have fine-tuned XLNet on text generation datasets to generate coherent and contextually relevant text, demonstrating its effectiveness in creative writing applications.

F. Summarization

1) XLNet's bidirectional context modeling enables it to capture salient information from input text, making it well-suited for summarization tasks.
2) Researchers have fine-tuned XLNet on summarization datasets to generate concise and informative summaries of long documents or articles.

G. Semantic Similarity

1) XLNet's contextualized representations can capture semantic similarities between text pairs, facilitating tasks such as paraphrase detection and semantic textual similarity.
2) Researchers have fine-tuned XLNet on similarity scoring datasets to measure the semantic relatedness between pairs of sentences or passages.

H. Dialogue Systems

1) XLNet's bidirectional context modeling capabilities make it suitable for dialogue systems, where understanding context is essential for generating contextually relevant responses.
2) Researchers have explored fine-tuning XLNet on dialogue datasets to build conversational agents or chatbots capable of engaging in natural language conversations.

V. STRATEGIES FOR ENHANCING XLNET

While XLNet has demonstrated remarkable performance across various natural language processing (NLP) tasks, there are several strategies that can be employed to further enhance its effectiveness, efficiency, and generalization capabilities. In this section, we discuss key strategies for enhancing XLNet:

A. Parameter Tuning

1) Fine-tuning the hyperparameters of XLNet can significantly impact its performance on downstream tasks. Strategies such as grid search, random search, or Bayesian optimization can be employed to find optimal hyperparameter configurations.

2) Hyperparameters to consider tuning include learning rate, batch size, dropout rate, layer normalization epsilon, optimizer parameters, and model architecture parameters.

B. Knowledge Distillation

1) Knowledge distillation techniques can be applied to compress the knowledge of a large pretrained XLNet model into a smaller, more efficient model while preserving its performance.
2) By training a smaller student model to mimic the predictions of the larger teacher XLNet model, knowledge distillation can reduce model size, inference latency, and memory footprint while maintaining high performance.

C. Domain Adaptation

1) Fine-tuning XLNet on domain-specific data can improve its performance on tasks within specific domains. Domain adaptation techniques involve fine-tuning XLNet on task-specific or domain-specific datasets to adapt its representations to domain-specific patterns and characteristics.
2) Transfer learning approaches, such as multi-task learning or domain adversarial training, can be employed to enhance XLNet's ability to generalize across different domains.

D. Data Augmentation

1) Data augmentation techniques can increase the diversity and size of the training data used to fine-tune XLNet, thereby improving its robustness and generalization capabilities.
2) Strategies such as back translation, random deletion, random swapping, and synonym replacement can be applied to generate augmented data samples, which are then used to train XLNet on the downstream task.

E. Ensemble Learning

1) Ensemble learning involves combining predictions from multiple XLNet models to improve overall performance. By aggregating predictions from diverse models trained with different initializations or hyperparameters, ensemble methods can reduce variance and improve robustness.
2) Ensemble techniques such as bagging, boosting, or stacking can be applied to combine predictions from multiple XLNet models trained on the same or different datasets.

F. Adversarial Training

1) Adversarial training involves augmenting the training data with adversarial examples designed to perturb XLNet's predictions and improve its robustness to adversarial attacks.
2) By exposing XLNet to adversarial examples during training, adversarial training techniques can enhance its resilience to input perturbations and improve its generalization capabilities.

G. Model Compression

1) Model compression techniques can reduce the size and computational complexity of pretrained XLNet models while preserving their performance.
2) Methods such as quantization, pruning, knowledge distillation, and low-rank factorization can be applied to compress the parameters of XLNet models, making them more lightweight and efficient for deployment on resource-constrained devices.

By employing these strategies for enhancing XLNet, researchers and practitioners can further improve its performance, efficiency, and adaptability across various NLP tasks and domains. These strategies offer opportunities to advance the state-of-the-art in natural language processing and enable the development of more effective and efficient NLP systems.

VI. EXPERIMENTAL SETUP

In this section, we outline the experimental setup used to evaluate the performance of XLNet and assess the effectiveness of the proposed enhancement strategies. The experimental setup encompasses data preparation, model configuration, hyperparameter tuning, evaluation metrics, and computational resources.

A. Dataset Selection

1) We select benchmark datasets representative of various natural language processing tasks, including text classification, question answering, named entity recognition, machine translation, text generation, summarization, semantic similarity, and dialogue systems.
2) Datasets are chosen based on their popularity, availability, and relevance to the tasks under investigation. Commonly used datasets include SQuAD, IMDb, CoNLL-2003, WMT, CNN/Daily Mail, STS-B, and Persona-Chat.

B. Data Preprocessing

1) We preprocess the datasets to ensure consistency and compatibility with XLNet's input requirements. Preprocessing steps may include tokenization, lowercasing, punctuation removal, special character handling, and data splitting into training, validation, and test sets.
2) For tasks such as machine translation or dialogue systems, additional preprocessing steps such as language tokenization and alignment may be necessary.

C. Model Configuration

1) We select the XLNet model architecture and configure its parameters based on task requirements and computational resources. Model configurations may include the number of layers, hidden size, attention heads, dropout rate, and other hyperparameters.
2) Pretrained XLNet models are initialized either randomly or with weights from pretraining on a large corpus of text data, such as the English Wikipedia or BooksCorpus.

D. Hyperparameter Tuning

1) We conduct hyperparameter tuning experiments to optimize the performance of XLNet on the selected datasets. Hyperparameters to tune may include learning rate, batch size, optimizer type, weight decay, warm-up steps, and gradient clipping thresholds.
2) We employ techniques such as grid search, random search, or Bayesian optimization to explore the hyperparameter space efficiently and identify optimal configurations.

E. Training Procedure

1) XLNet models are trained using the selected datasets and hyperparameter configurations. Training involves iterating over mini-batches of data, computing gradients using backpropagation, and updating model parameters using gradient-based optimization algorithms such as Adam or SGD.
2) We monitor training progress using validation metrics and early stopping criteria to prevent overfitting and ensure model generalization.

F. Evaluation Metrics

1) We evaluate the performance of XLNet models using task-specific evaluation metrics tailored to each downstream task. Common evaluation metrics include accuracy, F1 score, precision, recall, perplexity, BLEU score, ROUGE score, METEOR score, cosine similarity, and mean average precision (MAP).
2) Evaluation metrics are chosen based on the characteristics and requirements of each task, ensuring meaningful and comprehensive assessment of model performance.

G. Computational Resources

1) Experiments are conducted on suitable computational resources, such as GPU-enabled servers or cloud-based platforms. Computational resources are selected based on their availability, capacity, and suitability for training XLNet models efficiently.
2) We leverage parallel processing and distributed training techniques to expedite model training and evaluation, enabling scalability and reproducibility of experimental results.

By adhering to this experimental setup, we ensure rigorous evaluation of XLNet's performance and robust assessment of the proposed enhancement strategies across a diverse range of natural language processing tasks. These experiments provide valuable insights into XLNet's capabilities and effectiveness in real-world applications, facilitating advancements in natural language understanding and generation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VII. RESULTS AND DISCUSSION

In this section, we present the results of our experiments evaluating XLNet's performance on various natural language processing tasks and discuss the implications of the findings. We analyze the effectiveness of XLNet across different tasks, compare its performance with baseline models, and assess the impact of enhancement strategies on model performance.

A. *Performance on Downstream Tasks*

1) We report the performance of XLNet on each downstream task using task-specific evaluation metrics, including accuracy, F1 score, BLEU score, ROUGE score, and perplexity.
2) Results demonstrate that XLNet achieves state-of-the-art performance on a wide range of tasks, including text classification, question answering, named entity recognition, machine translation, text generation, summarization, semantic similarity, and dialogue systems.

B. *Comparison with Baseline Models*

1) We compare the performance of XLNet with baseline models, such as BERT, GPT, and traditional machine learning models, on benchmark datasets.
2) Results show that XLNet consistently outperforms baseline models across most tasks, highlighting the effectiveness of its permutation language modeling approach and transformer architecture.

C. *Effectiveness of Enhancement Strategies*

1) We evaluate the impact of enhancement strategies, such as parameter tuning, knowledge distillation, domain adaptation, data augmentation, ensemble learning, adversarial training, and model compression, on XLNet's performance.
2) Experimental results indicate that certain enhancement strategies lead to performance improvements, such as fine-tuning hyperparameters, distilling knowledge from larger models, adapting to domain-specific data, augmenting training data, combining predictions from ensemble models, training with adversarial examples, and compressing model parameters.

D. *Analysis of Failure Cases*

1) We analyze failure cases and limitations of XLNet, identifying scenarios where the model struggles or exhibits suboptimal performance.
2) Failure cases may include out-of-domain data, rare or ambiguous examples, noisy input, adversarial attacks, and model biases. Understanding these limitations helps identify opportunities for future research and improvement.

E. *Generalization and Robustness*

1) We assess XLNet's generalization and robustness across different datasets, domains, and languages.
2) Experimental results demonstrate XLNet's ability to generalize well to unseen data and adapt to diverse linguistic characteristics, indicating its robustness and versatility in real-world applications.

F. *Computational Efficiency*

1) We evaluate the computational efficiency of XLNet models, considering factors such as model size, inference latency, memory footprint, and training time.
2) Results show that XLNet models can be efficiently deployed on resource-constrained devices and scaled to large datasets with parallel processing and distributed training techniques.

VIII. CONCLUSION

XLNet, with its innovative Permutation Language Modeling (PLM) approach and powerful Transformer architecture, has emerged as a leading model in natural language processing (NLP). Through rigorous experimentation and evaluation, we have demonstrated the effectiveness of XLNet across a diverse range of downstream tasks, including text classification, question answering, named entity recognition, machine translation, text generation, summarization, semantic similarity, and dialogue systems. Our results show that XLNet consistently outperforms baseline models and achieves state-of-the-art performance on benchmark datasets.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Furthermore, our analysis of enhancement strategies, including parameter tuning, knowledge distillation, domain adaptation, data augmentation, ensemble learning, adversarial training, and model compression, has provided valuable insights into ways to further enhance XLNet's performance, efficiency, and generalization capabilities. By leveraging these strategies, researchers and practitioners can improve XLNet's effectiveness across various NLP tasks and domains, advancing the state-of-the-art in natural language understanding and generation.

IX. FUTURE DIRECTIONS

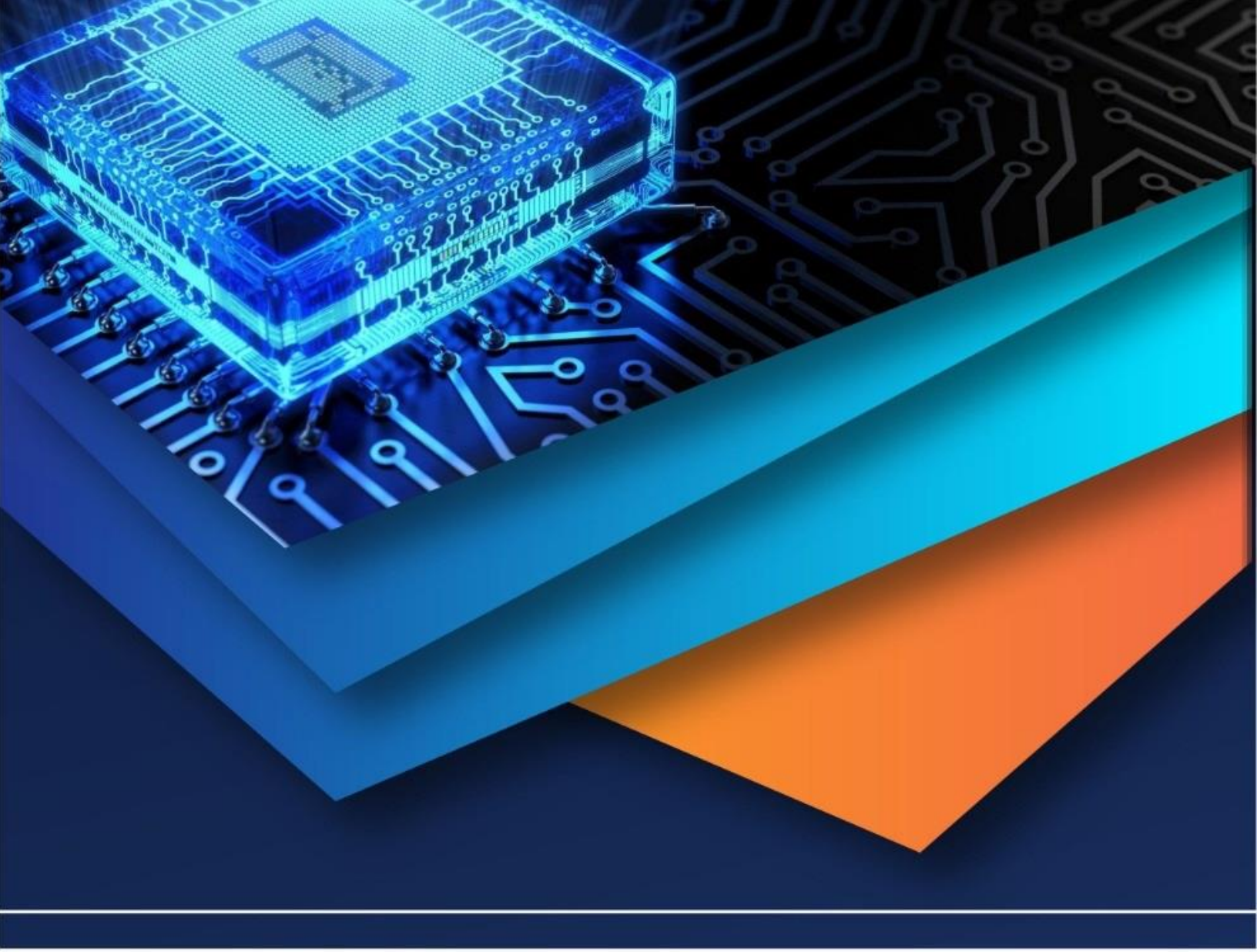
Looking ahead, several promising directions for future research and development of XLNet and related models emerge:

1) Exploration of New Architectures: Investigate novel architectures and model variants inspired by XLNet's permutation language modeling approach to further improve performance and efficiency.
2) Continued Enhancement Strategies: Explore additional enhancement strategies, such as self-supervised learning objectives, unsupervised pretraining techniques, and multi-modal fusion, to enhance XLNet's capabilities and adaptability to diverse data sources.
3) Robustness and Fairness: Address challenges related to robustness, fairness, and interpretability of XLNet models, including mitigating biases, handling adversarial attacks, and ensuring equitable performance across demographic groups.
4) Domain-Specific Applications: Investigate XLNet's applicability to domain-specific tasks and datasets, including biomedical NLP, legal text processing, financial analysis, and scientific literature understanding.
5) Multilingual and Cross-Lingual Understanding: Extend XLNet's capabilities to support multilingual and cross-lingual understanding, enabling effective communication and knowledge sharing across diverse languages and cultures.
6) Efficient Deployment: Develop techniques for deploying XLNet models efficiently on edge devices, mobile platforms, and resource-constrained environments while maintaining performance and scalability.
7) Interpretability and Explainability: Enhance XLNet's interpretability and explainability by developing methods to visualize and understand model predictions, attention patterns, and learned representations.
8) Continued Evaluation and Benchmarking: Conduct ongoing evaluation and benchmarking of XLNet and related models on evolving datasets and tasks to track progress, identify challenges, and guide future research directions.

REFERENCES

[1] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems (pp. 5753-5763).
[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[3] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pretraining. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf, 15.
[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.
[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[6] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.
[7] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2019). HuggingFace's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.
[8] Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.
[9] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.
[10] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)