Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

A Deep Dive in to Neural Models in NLP

Aisheek Mazumder
Computer Science and Engineering
Chandigarh University
Cha ndigarh, India

Kumar Sanu
Computer Science and Engineering
Chandigarh University
Chandigarh, India

Ayush Kumar
Computer Science and Engineering
Chandigarh University
Chandigarh, India

Prabhat Kumar
Computer Science and Engineering
Chandigarh University
Chandigarh, India

Aryan Chauhan
Computer Science and Engineering
Chandigarh University
Chandigarh, India

Er. Simran Kaur Birdi
Computer Science and Engineering
Chandigarh University
Chandigarh, India

***Abstract*-Neural models have transformed the landscape of Natural Language Processing (NLP), enabling significant advancements in tasks such as machine translation, sentiment analysis, and question answering. However, their complexity and opacity create challenges in understanding their internal mechanisms and decision-making processes. This paper delves into interpretability and visualization techniques designed for neural models in NLP, emphasizing the growing need for methods that enable deeper comprehension of model behavior. Through a review of attention mechanisms, saliency maps, and probing classifiers, this work explores how these techniques help uncover the layers of neural models, offering insights into their predictions and outputs. Additionally, case studies on Long Short-Term Memory (LSTM) networks and Transformer models demonstrate how visualization has been applied to reveal the inner workings of these architectures. Finally, the paper discusses the implications of interpretability for future research and the potential for the development of new tools that could facilitate more transparent and accountable NLP systems.**

***Keywords*: Neural Networks(NN), Natural Language Processing, Long Short-Term Memory(LTSM), BERT, GPT**

I. INTRODUCTION

The rise of deep learning has revolutionized the field of Natural Language Processing (NLP). Prior to neural networks, NLP methods largely depended on statistical models and feature-based machine learning, which required labor-intensive manual feature engineering. These traditional models struggled with complex language tasks due to their reliance on hand-crafted features and their inability to capture the rich, intricate structures of human language. The advent of neural networks, particularly models like Long Short-Term Memory (LSTM) networks and, more recently, Transformer-based architectures such as BERT and GPT, has significantly advanced the state of NLP, delivering remarkable performance across tasks like machine translation, question answering, and sentiment analysis.

Neural networks have changed the landscape of NLP by automatically learning high-level representations of language from vast amounts of data. Unlike previous models, which had to be meticulously designed to recognize specific linguistic patterns, neural models can learn these patterns on their own. This ability has allowed for significant breakthroughs, particularly with the development of Transformer models. Transformers, with their attention mechanisms, have set new benchmarks in NLP, enabling much deeper understanding and generation of text. As a result, neural networks have replaced traditional approaches in many applications, achieving unprecedented results in areas like text classification, named entity recognition, and summarization.

However, this success comes with a downside: the "black box" nature of neural models. Unlike simpler, more transparent models, neural networks are often opaque, making it difficult to explain how they arrive at their decisions. This lack of interpretability has become a critical issue, especially as NLP systems are being deployed in high-stakes fields such as healthcare, finance, and law, where understanding the rationale behind a model's output is essential. Interpretability refers to the ability of a human to comprehend and trust the decisions made by a machine learning model, and in the context of NLP, it is vital for ensuring responsible and ethical use of AI systems. As a result, researchers have increasingly focused on developing methods to improve the interpretability of neural models in NLP. Several techniques have been proposed to help understand and explain these complex models. One prominent approach is the use of saliency maps, which highlight the parts of the input that are most influential in a model's decision-making process. Another method is to analyze attention mechanisms, particularly in Transformer models, where attention weights can provide insights into which words or phrases the model is focusing on when making predictions. Probing classifiers are also used to test what specific linguistic knowledge (such as syntax or semantics) is encoded in the model's hidden layers.

By leveraging these interpretability techniques, researchers can gain a better understanding of the inner workings of neural models. This knowledge not only helps in diagnosing and correcting errors in NLP systems but also fosters greater trust in their outputs, particularly in critical real-world applications.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

Enhancing interpretability ensures that AI systems are both effective and accountable, paving the way for safer, more reliable deployment of neural NLP models across various domains.

II. BACKGROUND ON NEURAL MODELS

This article [1] explores advancements in text classification, a crucial task in Natural Language Processing (NLP) that involves categorizing text into predefined categories. The paper presents a thorough review of traditional and modern approaches to text classification, emphasizing the shift from earlier statistical and rule-based methods to deep learning-based techniques. Initially, Wang reviews conventional methods like Bag of Words (BoW) and TF-IDF, which rely on manually engineered features. These methods, while foundational, are limited by their inability to capture semantic relationships between words and contextual nuances within text. Support Vector Machines (SVMs) and Naive Bayes classifiers are highlighted as widely used algorithms before the rise of deep learning. The study then transitions to discussing modern deep learning methods, focusing on neural network-based models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RVNNs), and Long Short-Term Memory (LSTM) networks. Wang emphasizes how these models can automatically learn feature representations from raw text, significantly improving classification performance. In this paper [2] it reviews existing sentiment analysis approaches and propose a hybrid model that integrates Recurrent Neural Networks (RVNNs) with Enhanced Multi-Head Self-Attention mechanisms. The authors emphasize that attention mechanisms, especially Multi-Head Self-Attention used in Transformer models, have significantly improved performance by enabling models to focus on different parts of the text simultaneously. However, they note that while Transformers excel at capturing global context, RVNNs are still effective in modeling sequential patterns, thus motivating their hybrid approach that combines RVNN structures with enhanced self-attention. The study [14] provide a comprehensive review of discourse parsing techniques, emphasizing the advancements brought by neural approaches and large-scale pretraining. The literature review traces the development of discourse parsing from early rule-based systems to modern neural methods, focusing on the challenges of capturing both contextual and structural information in discourse. Initially, the authors discuss traditional discourse parsing methods, which relied on manual rules and feature-based machine learning models. These methods, while effective in simple cases, struggled with capturing complex discourse structures and contextual nuances. Rhetorical Structure Theory (RST), a widely used framework for representing discourse, was foundational in early approaches, but these models required extensive feature engineering and were often brittle. The work [15] provide a thorough review of existing techniques for extractive summarization, particularly for long documents, before introducing their novel graph-based approach. The literature review explores the evolution of summarization methods from traditional machine learning approaches to more recent neural network models. It emphasize the potential of graph-based approaches to overcome these limitations by representing documents as graphs, where nodes can represent sentences, passages, or even key entities, and edges capture relationships between them. The review highlights how Graph Neural Networks (GNNs) have been increasingly adopted for document-level tasks as they excel in modeling complex dependencies across different sections of a text. The authors specifically point out the limitations of existing graph-based models, which often treat all relationships uniformly, ignoring the heterogeneity of document structures. This motivates the introduction of their Heterogeneous Graph Neural Network (HeterGraphLongSum) approach, which distinguishes between different types of relationships (e.g., passage-to-passage or passage-to-entity) and uses passage-level aggregation to summarize long documents more effectively.

III. VISUALISING TECHNIQUES FOR NEURAL MODELS

Visualization techniques for neural models in Natural Language Processing (NLP) are essential for understanding how these complex models work, particularly in cases where interpretability is critical. These techniques help researchers and practitioners gain insights into what parts of the input data influence the model's decisions and how the model processes information. Here are some key visualization techniques used in neural NLP models.

A. Saliency Maps

Saliency maps are powerful visualization tools used to understand which parts of an input contribute most significantly to a model's predictions. In the context of Natural Language Processing (NLP), saliency maps highlight the importance of specific words or phrases within a text. The technique involves calculating the gradients of the model's output with respect to the input features—essentially measuring how changes in input words affect the final prediction. This is often done using backpropagation, where the model determines the sensitivity of the output score to each input word.

By creating saliency maps, researchers can visually represent the areas of the text that have the highest influence on the model's decision, thereby providing insight into the model's internal workings. For example, in sentiment analysis, a saliency map might show that words like "excellent" or "terrible" receive high scores, indicating their significant impact on the sentiment classification. These maps are particularly useful for interpreting complex models, as they help users understand not just what the model predicts, but also why it makes certain predictions. This transparency is crucial for applications in sensitive areas, such as healthcare or finance, where trust in AI systems is paramount.

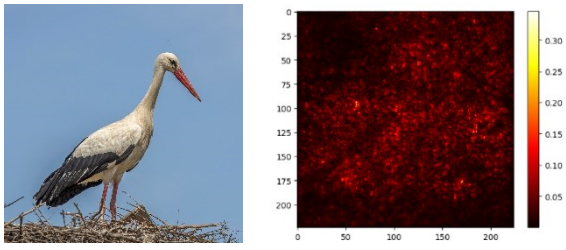

Fig. 1. a) Input images b) image's Saliency Map

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

B. Attention Mechanisms

Attention mechanisms have become a cornerstone of modern neural networks, particularly in Transformer architectures like BERT and GPT. The primary function of attention is to allow models to focus selectively on different parts of the input when making predictions. Unlike traditional sequential models that process input tokens in order, attention mechanisms evaluate the relationships between all tokens simultaneously, assigning different weights to them based on their relevance.

This mechanism works by calculating attention scores, which determine how much focus the model should place on each token when generating an output. In NLP, attention visualizations, often presented as heatmaps, reveal which words the model prioritizes during tasks like translation or summarization. For instance, in machine translation, an attention heatmap can illustrate how a model aligns source-language words with their target-language counterparts, providing insights into the model's understanding of linguistic structure. By visualizing attention weights, researchers can better comprehend how the model navigates complex contexts, leading to more interpretable and trustworthy predictions.

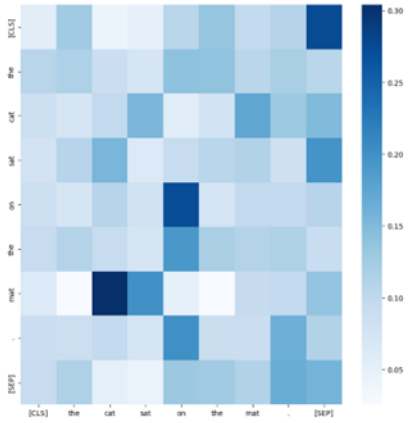

Fig. 2. Attention in NLP using Transformers

C. Layer-wise Relevance Propagation (LRP)

Layer-wise Relevance Propagation (LRP) is a sophisticated technique designed to enhance the interpretability of neural networks by elucidating how input features contribute to a model's output. The essence of LRP lies in its ability to trace back the decision-making process through the layers of the neural network. This technique assigns relevance scores to each input feature by decomposing the output prediction into contributions from the input features, effectively creating a relevance distribution across the input.

In the context of NLP, LRP helps reveal which specific words or phrases are most influential in driving a particular prediction. For instance, in a sentiment classification task, LRP can identify that certain emotionally charged words are primarily responsible for the model's positive or negative classification. By providing a clear attribution of relevance, LRP allows practitioners to assess the fairness and accountability of their models. Furthermore, LRP can serve as a diagnostic tool to identify biases or errors in model predictions, thereby supporting the development of more robust and ethically aligned AI systems.

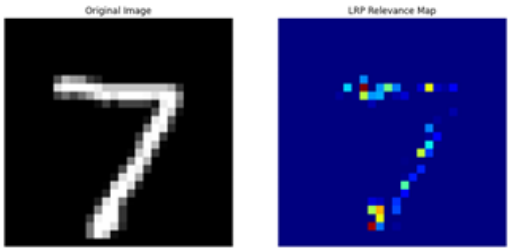

Fig. 3. Original image vs LRP map

D. t-SNE and PCA for Embedding Visualization

t-Distributed Stochastic Neighbour Embedding (t-SNE) and Principal Component Analysis (PCA) are dimensionality reduction techniques widely employed to visualize high-dimensional data, such as word or sentence embeddings generated by neural models. These techniques are particularly useful in NLP, where embedding spaces can have hundreds of dimensions, making it challenging to interpret the relationships between words or documents. PCA works by identifying the principal components of the data—essentially the directions of maximum variance—and projecting the data into a lower-dimensional space while retaining as much information as possible. This helps in understanding the overall structure of the data and identifying clusters of similar items. t-SNE, on the other hand, is specifically designed for visualizing high-dimensional data by preserving local relationships. It converts high-dimensional distances into probabilities and aims to minimize the divergence between the probability distributions of the original and low-dimensional spaces. As a result, t-SNE excels in revealing intricate structures and clusters within the data, making it particularly effective for visualizing word embeddings, where semantically similar words cluster together in the reduced space.

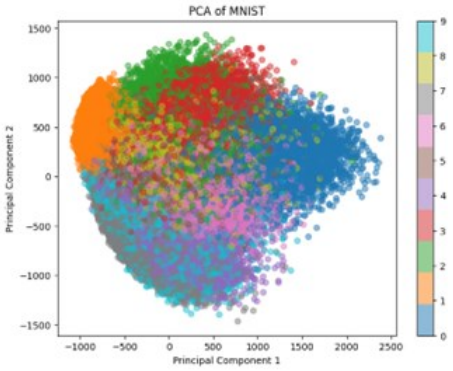

Fig. 4. PCA of MNSIT Dataset

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

IV. MODELS IN NLP

A. Long short-term memory (LSTM)

Long Short-Term Memory (LSTM) networks, a specialized type of Recurrent Neural Network (RVNN), have emerged as one of the most influential architectures in the field of deep learning, particularly for sequential and time-series data. Developed by Sepp Hochreiter and Jürgen Schmidhuber in 1997, LSTMs were designed to address one of the fundamental challenges of traditional RVNNs: their difficulty in capturing long-range dependencies within sequences. The architecture of LSTMs enables them to effectively remember information for extended periods, making them particularly well-suited for tasks involving sequences where context is critical. Before the advent of LSTM networks, RVNNs were widely used for processing sequential data due to their ability to maintain hidden states that can capture information from previous time steps. However, traditional RVNNs faced significant limitations due to the vanishing and exploding gradient problems. During backpropagation, gradients can diminish exponentially for long sequences, rendering the network unable to learn long-term dependencies effectively. Conversely, they can also grow excessively, leading to unstable updates. This made it challenging for RVNNs to learn patterns that span across many time steps, which is often crucial in applications such as natural language processing, speech recognition, and time-series forecasting.

The architecture of LSTM networks is characterized by a unique structure that includes memory cells and three key gating mechanisms: the input gate, the forget gate, and the output gate. These components work together to regulate the flow of information, enabling LSTMs to retain relevant information over long sequences while discarding irrelevant data.

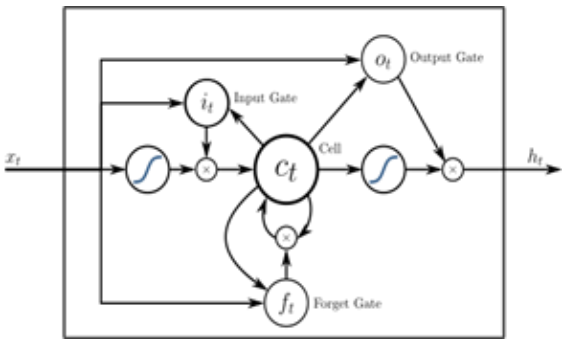

Fig. 5. Architecture of LSTM networks

Memory Cells: The core of the LSTM architecture is the memory cell, which serves as a storage unit for information. It maintains the cell state over time, allowing the network to carry information forward through the sequence.

Gating Mechanisms:

- Input Gate: The input gate determines how much of the new information from the current time step should be added to the cell state. It uses a sigmoid activation function to produce values between 0 and 1, effectively controlling the extent of information inclusion.
- Forget Gate: The forget gate regulates which information in the cell state should be discarded. It assesses the importance of existing information and decides what to forget based on the current input and the previous hidden state.
- Output Gate: The output gate determines what part of the cell state should be outputted to the next layer or time step. It filters the cell state and passes the relevant information forward while also controlling the hidden state.

These gates allow LSTMs to maintain a selective memory, thus overcoming the limitations of traditional RVNNs and enabling the network to learn temporal patterns more effectively.

B. Recursive neural network (RvNN)

Recursive Neural Networks (RvNNs) represent a significant advancement in the field of deep learning, particularly in the domain of structured data. While traditional feedforward neural networks and convolutional neural networks (CNNs) excel in handling fixed-size inputs such as images, RvNNs are designed to process data with an inherent recursive structure. This makes them particularly well-suited for tasks involving hierarchical or tree-like relationships, such as natural language processing, parsing, and even computer vision tasks where spatial hierarchies are present. Recursive structures are characterized by the repetition of similar patterns or elements, where a structure can be defined in terms of itself. This is common in natural language processing, where sentences can be represented as parse trees, with words serving as the leaves and phrases or clauses as internal nodes. In this context, RvNNs provide a powerful mechanism for modeling such relationships, allowing them to capture the intricacies of language and meaning.

Recursive Neural Networks utilize the idea of applying the same neural network recursively to the inputs that are structured hierarchically. For instance, in a binary tree structure, the RvNN would combine the representations of child nodes to generate a representation for the parent node. This recursive approach allows the network to learn a hierarchy of representations, enabling it to capture complex relationships and dependencies within the data.

The architecture of an RvNN is based on the principle of recursive composition, where the same neural network is applied at each level of the hierarchy. Unlike traditional RvNNs that process sequences in a linear fashion, recursive neural networks process structured data by following the tree structure.

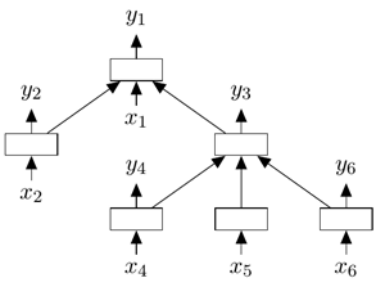

Fig. 6. Architecture Recursive neural network

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

- Input Representation: Each node in the tree corresponds to a data point, such as a word or phrase. The input to the network is typically a vector representation of these points, often obtained through techniques such as word embeddings.
- Composition Function: The core of the RvNN architecture is the composition function, which combines the representations of child nodes to form the representation of a parent node. This function can be a simple operation like addition or multiplication, but it is usually implemented as a neural network layer (e.g., a feedforward network) that learns to weigh and combine the inputs.
- Recursive Application: The recursive application of the composition function continues up the tree until the root node is reached, resulting in a holistic representation of the entire structure. This final representation can then be used for various tasks, such as classification, regression, or further processing.

Recursive Neural Networks represent a powerful and flexible approach to modeling structured data, particularly in domains where relationships are inherently hierarchical. Their ability to capture complex dependencies and relationships makes them valuable for a wide range of applications, from natural language processing to computer vision and beyond. Through ongoing advancements, Recursive Neural Networks are poised to continue shaping the landscape of machine learning and artificial intelligence.

V. NLP MODELS VISUALISATION

A. Steps for Visualizing Recursive neural network (RvNN)

- Understand the Structure: Before visualizing, familiarize yourself with the hierarchical structure of the data you want to model, such as parse trees in NLP. This understanding is crucial for creating meaningful visualizations.
- Use Visualization Tools: Tools like Graphviz, TensorBoard, or custom Python scripts using libraries like Matplotlib or Seaborn can be helpful for creating visual representations of RVNNs.
- Generate Parse Trees: If your data is in a tree structure (like sentences in NLP), generate the parse trees using tools like NLTK or SpaCy. Visualize these trees to show the hierarchical relationships.
- Display the Composition Function: Visualize how the composition function combines child node representations into parent nodes. Use arrows to indicate the flow of information from child nodes to the parent node.
- Highlight the Recursive Nature: Illustrate the recursive application of the network. Use nested boxes or layers to represent the repeated application of the composition function.
- Show Activation Maps: Visualize the activation values of the nodes as the input propagates through the network. You can use heatmaps to indicate which nodes are most activated during specific inputs.
- Visualize Outputs: If applicable, show how the final representation (root node) translates into output. You can visualize the final decision or classification made by the network.
- Interactive Visualizations: For a more dynamic understanding, use tools like TensorBoard or D3.js for interactive visualizations. This allows you to explore how different inputs affect the network's behavior in real-time.

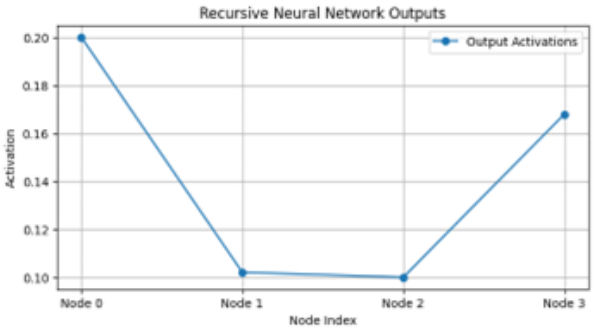

Fig. 6. Recurrent neural network outputs

B. Steps for Visualizing Long short-term memory (LSTM) Understand LSTM Components: Familiarize yourself with the key components of LSTMs: memory cells, input gates, forget gates, and output gates. Understanding these elements is essential for effective visualization.

- Use Visualization Libraries: Utilize libraries like Keras for model visualization, which provides built-in functions to plot the architecture of LSTM networks. Other tools like Graphviz or Matplotlib can also be used for custom visualizations.
- Plot the LSTM Architecture: Create a diagram of the LSTM architecture showing the input layer, LSTM layers, and output layer. Clearly label the memory cell and the gating mechanisms.
- Visualize Gates: Illustrate the flow of data through the different gates. You can create flowcharts or diagrams to depict how information is added, retained, or forgotten at each time step.
- Show Cell States: Visualize the cell state across different time steps. Use line graphs to show how the cell state evolves over time, highlighting important moments where the state changes significantly.
- Visualize Hidden States: Similar to cell states, visualize the hidden states throughout the sequence. Heatmaps or line graphs can show how the hidden state responds to inputs over time.
- Activation Functions: Illustrate the activation functions used in each gate (sigmoid and tanh) and show how they transform inputs. You can plot the function curves alongside the input data to provide a clearer picture.
- Layer Outputs: For each time step, visualize the outputs of the LSTM layers. This can be done using bar charts or line graphs to indicate how the model's predictions change with different inputs.
- Interactive Visualizations: Use interactive tools such as TensorBoard to visualize training processes, including loss, accuracy, and the internal state of the LSTM over epochs. This interactivity provides deeper insights into the learning dynamics.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

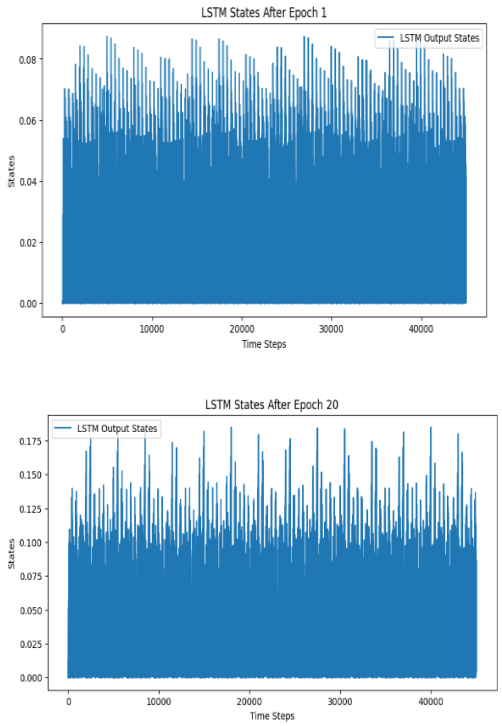

Fig. 7. LTSM states at various epochs

VI. RESULTS

Comparative visualizations of Recursive Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks highlight their structural and operational differences, as well as their performance characteristics. RNN visualizations typically depict hierarchical structures, such as parse trees, showing how child nodes contribute to parent representations through recursive composition. In contrast, LSTM visualizations emphasize layered architectures, detailing the roles of input, forget, and output gates alongside memory cells, illustrating how they manage information flow. Performance metrics can be displayed through bar charts and loss/accuracy plots, demonstrating LSTMs' superior ability to capture long-range dependencies compared to traditional RNNs. Side-by-side comparisons of these elements, including architectural diagrams and task-specific results, provide clear insights into the advantages and applications of each model type in handling sequential and structured data.

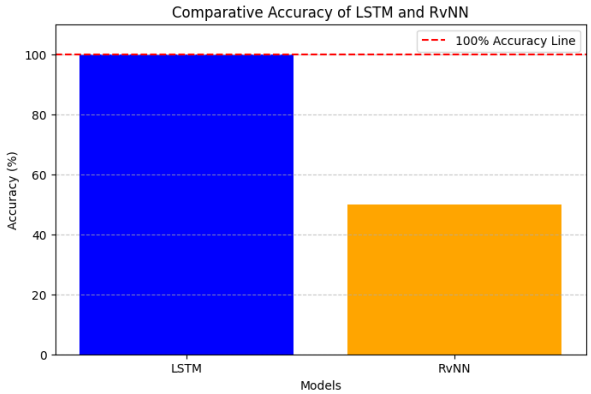

Fig. 8. Graph of accuracy of LTSM and RvNN

VIII. CONCLUSION AND FUTURE SCOPE

Neural models have revolutionized the field of Natural Language Processing (NLP), driving substantial advancements in a wide range of applications such as machine translation, sentiment analysis, question answering, and text summarization. Traditional methods that relied on handcrafted features and statistical techniques have been largely supplanted by deep learning architectures, which automatically learn rich representations from vast amounts of data. This shift has enabled state-of-the-art performance across various NLP tasks, with models like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer architectures fundamentally changing how machines understand and generate human language. The introduction of LSTMs addressed the limitations of standard RNNs, particularly the challenges associated with vanishing gradients, thus enabling models to learn from longer sequences. More recently, Transformer models, characterized by their attention mechanisms, have set new benchmarks by allowing parallelization and better capturing dependencies across entire sequences without the limitations of recurrent processing. Models such as BERT and GPT have further pushed the boundaries, showcasing the effectiveness of pretraining on large corpora followed by fine-tuning for specific tasks. The future scope of neural models in Natural Language Processing (NLP) is brimming with opportunities for innovation and improvement. Key areas of focus include enhancing model interpretability to build trust in high-stakes applications, developing more efficient and scalable architectures that operate effectively in resource-constrained environments, and addressing the needs of low-resource languages through transfer and few-shot learning techniques. Additionally, researchers must tackle ethical considerations by identifying and mitigating biases inherent in training data. Advancements in cross-modal learning, continual learning, and sophisticated dialogue systems will further enrich NLP capabilities. The exploration of unsupervised and self-supervised learning paradigms can reduce reliance on labeled data, while domain-specific adaptations will enhance model performance in specialized fields like healthcare and finance. Ultimately, fostering human-AI collaboration will be crucial for creating effective systems that enhance human-computer interaction while ensuring ethical oversight.

ACKNOWLEDGEMENT

We would like to thank Ms Simran Kaur Birdi for necessary discussion and problem solving for doing this research work.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 10, October 2024

REFERENCES

[1] Wang, M., 2023. Research on Text Classification Method Based on NLP. Advances in Computer, Signals and Systems, 7(2), pp.93-100.

[2]] Leng, X.L., Miao, X.A. and Liu, T., 2021. Using recurrent neural network structure with enhanced multi-head self-attention for sentiment analysis. Multimedia Tools and Applications, 80, pp.12581-12600

[3] Jing, K. and Xu, J., 2019. A survey on neural network language models. arXiv preprint arXiv:1906.03591.

[4] Cui, K., Hao, R., Huang, Y., Li, J. and Song, Y., 2023. A novel convolutional neural networks for stock trading based on DDQN algorithm. IEEE Access, 11, pp.32308-32318.

[5] Guz, G., Huber, P. and Carenini, G., 2020. Unleashing the Power of Neural Discourse Parsers--A Context and Structure Aware Approach Using Large Scale Pretraining. arXiv preprint arXiv:2011.03203.

[6] Zaytsev, A., Natekin, A., Vorsin, E., Smirnov, V., Smirnov, G., Sidorshin, O., Senin, A., Dudin, A. and Berestnev, D., 2023. Designing an attack-defense game: how to increase robustness of financial transaction models via a competition. arXiv preprint arXiv:2308.11406.

[7] Phan, T.A., Nguyen, N.D.N. and Bui, K.H.N., 2022, October. Hetergraphlongsum: heterogeneous graph neural network with passage aggregation for extractive long document summarization. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 6248-6258).

[8] Shao, N., Cui, Y., Liu, T., Wang, S. and Hu, G., 2020. Is graph structure necessary for multi-hop question answering?. arXiv preprint arXiv:2004.03096.

[9] Hao, Z., 2024. Comparative Analysis of Transformer Integration in U-net Networks for Enhanced Medical Image Segmentation. Highlights in Science, Engineering and Technology, 94, pp.333-340.

[10] Bac, N.B.T. and Van Vinh, N., 2023, December. Improving Vietnamese Question-Answering system with Data Augmentation and Optimization. In 2023 RIVF International Conference on Computing and Communication Technologies (RIVF) (pp. 401-406). IEEE.

[11] Martin, A., Pedersen, T. and D'Souza, J., 2022, November. NLPSharedTasks: A Corpus of Shared Task Overview Papers in Natural Language Processing Domains. In Proceedings of the first Workshop on Information Extraction from Scientific Publications (pp. 105-120).

[12] Singh, G., Mittal, N. and Chouhan, S.S., 2022. A Systematic Review of Deep Learning Approaches for Natural Language Processing in Battery Materials Domain. IETE Technical Review, 39(5), pp.1046-1057.

[13] Cing, D.L. and Soe, K.M., 2020. Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar Language. International Journal of Electrical and Computer Engineering, 10(2), p.2023.

[14] Simov, K., Koprinkova-Hristova, P., Popov, A. and Osenova, P., 2023. A reservoir computing approach to word sense disambiguation. Cognitive Computation, pp.1-10.

[15] Soni, V.K., Gopalani, D. and Govil, M.C., 2019. Resurgence of Deep Learning: Genesis of Word Embedding. In Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS 2017, Volume 1 (pp. 129-139). Springer Singapore.

[16] Chan, S.W.K. and Franklin, J., 1994, April. A neural network model for acquisition of semantic structures. In Proceedings of ICSIPNN'94. International Conference on Speech, Image Processing and Neural Networks (pp. 221-224). IEEE.

[17] Rabin, M.R.I., Hussain, A., Suneja, S. and Alipour, M.A., 2023, May. Study of distractors in neural models of code. In 2023 IEEE/ACM International Workshop on Interpretability and Robustness in Neural Software Engineering (InteNSE) (pp. 1-7). IEEE.

[18] Li, J., Chen, X., Hovy, E. and Jurafsky, D., 2015. Visualizing and understanding neural models in NLP. arXiv preprint arXiv:1506.01066.