(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Predictive Analytics

PROJECT REPORT

ON

" **Comparative Analysis of LSTM, Attention, and Transformers for Headline Generation**"

Submitted By:

- Om Bhutkar – *202201040111*
- Bhavesh Bagul - *202201040111*
- Yash Mali – 202201040093

Batch: *DL - 2*

Guidance By :

Prof. Dippti Ghusse

Academic Year: 2024-25

Title:

Comparative Analysis of LSTM, Attention Mechanisms, and Transformers for News Headline Generation

Abstract :-

This project focuses on **automated news headline generation** using deep learning techniques.

- Three models are compared:
  - **LSTM/GRU without Attention**
  - **LSTM with Attention (Bahdanau/Luong)**
  - **Transformer (Self-Attention)**
- A real-world **news summary dataset** is used for training and evaluation.
- Performance is measured using **BLEU, ROUGE, and METEOR** scores.
- Models are also compared on **training time, inference speed, and complexity**.
- Results show:
  - Attention and Transformer models outperform basic LSTM/GRU.
  - Transformers give the best accuracy and fluency, with higher resource demands.
- The study helps in selecting suitable models based on **performance vs. efficiency trade-offs**

2. Introduction

- Headline generation is a key task in NLP that condenses news content into a short, meaningful title.
- Headlines help readers quickly grasp the essence of an article.
- Automatic headline generation saves time and enables real-time content delivery.
- Challenges include:
  - Semantic understanding of the full text
  - Grammatical correctness and fluency
  - Maintaining relevance and brevity
- The project compares different neural architectures:
  - LSTM/GRU (basic sequence models)
  - Attention-based models (Bahdanau, Luong)
  - Transformer-based models
- Aim: Evaluate each model's performance in generating accurate and fluent headlines.

3. Literature Review

"Sequence-to-Sequence Abstractive Text Summarization Model for Headline Generation with Attention"

Paper Overview

This paper presents a **sequence-to-sequence (Seq2Seq) architecture** augmented with an **attention mechanism** for generating news headlines. The model aims to generate concise and relevant summaries (headlines) from full-length news articles using deep learning-based natural language generation techniques.

Model Insights

- **Base Architecture**: The system uses the traditional **encoder-decoder** framework:
  - The **encoder** converts the input news article into a fixed-size context vector.
  - The **decoder** generates the headline word-by-word from this context.
- **Attention Mechanism**:
  - Instead of relying solely on the final encoder state, attention allows the decoder to "focus" on different parts of the input during each step of decoding.
  - It dynamically weights encoder outputs based on relevance to the current decoder state.
  - This helps capture long-range dependencies and semantic relationships better than vanilla Seq2Seq.
- **Training**:
  - Trained on a large dataset of news articles with paired headlines.
  - Loss function: **categorical cross-entropy**.
  - Optimizer: typically **Adam** for faster convergence (though exact optimizer isn't specified in the text preview).
- **Evaluation Metrics**:
  - Uses common NLP summarization metrics like **BLEU**, **ROUGE**, and **METEOR** to evaluate the quality of generated headlines.

Deep Understanding of Attention / Transformer Concepts

Although the paper implements a **Seq2Seq with attention**, not a full Transformer, it aligns with the foundational concepts of attention used in Transformer models:

Concept	Seq2Seq + Attention	Transformer
Attention Type	Bahdanau or Luong (soft attention over encoder)	Self-attention in encoder and decoder layers
Parallelism	Limited (depends on RNN structure)	Fully parallelizable
Long-Range Dependency	Handled better than plain RNN, but not as powerful	Excellent, via positional encodings + attention
Architecture	RNN-based (LSTM/GRU)	No RNNs; fully attention-based

Strengths of the Proposed Model

- Generates **context-aware headlines** using attention.
- **More interpretable** due to the attention weights.
- **Improves over traditional Seq2Seq** in fluency and relevance of headlines.
- Suitable for **abstractive summarization**, not just extractive.

Limitations & Future Work

- Limited scalability due to reliance on RNNs.
- Transformer-based architectures may outperform it on larger corpora.
- Future work could explore **hybrid models or full Transformer architectures** for better generalization and training efficiency.

4. Dataset Description

- The dataset used for this project is the **News Headline Generation Dataset**, available on Kaggle (click here).
- It contains **news articles and their corresponding headlines**, designed specifically for text summarization tasks.
- Dataset details:
  - Over **11,000 rows** of news content and headlines
  - Two main columns: content_text (article) and generated_headline (headline)
- Preprocessing steps performed:
  - **Lowercasing** all text
  - **Removing punctuation and special characters**
  - **Stripping extra spaces**
  - **Tokenization** using Keras' Tokenizer

- **Padding** sequences for model input compatibility

5. Methodology

◆ A. LSTM/GRU (Without Attention)

- Utilizes an **Encoder-Decoder architecture** built with LSTM or GRU units.
- **Encoder** processes the input sequence into a fixed-size **context vector**.
- **Decoder** generates the headline word by word using this context vector.
- **Limitations**:
  - Struggles with **long-range dependencies** in longer inputs.
  - Context vector acts as a **bottleneck**, losing important details from earlier parts of the input sequence.

◆ B. LSTM with Attention (Bahdanau / Luong)

- Integrates an **Attention Mechanism** (Bahdanau or Luong) into the encoder-decoder model.
- Instead of relying on a single context vector, attention allows the decoder to:
  - **Dynamically focus** on relevant parts of the input at each decoding step.
  - Improve **semantic alignment** between input and output.
- This helps overcome the bottleneck and enhances output fluency and relevance.

◆ C. Transformer (Self-Attention Based)

- Uses **Self-Attention** instead of recurrence to process input and output sequences.
- Employs **multi-head attention** and **positional encoding** to retain word order.
- **Benefits**:
  - Enables **parallel processing** of sequences, making training faster.
  - Handles **long sequences efficiently** and captures complex relationships.
- This architecture is the backbone of state-of-the-art NLP models like BERT and GPT.

9. Discussion

Best Performing Model:
The Transformer-based model outperforms LSTM/GRU and LSTM with Attention. It provides superior accuracy, fluency, and ability to handle long-range dependencies, making it ideal for tasks like news headline generation.

Trade-offs:

- **LSTM/GRU (without Attention)**: Simpler and faster to train but struggles with long-range dependencies.

- **LSTM with Attention**: Improves performance over vanilla LSTM by addressing long-range dependencies and enhancing output fluency. However, it's still slower than Transformer models.
- **Transformer**: Provides the best accuracy and fluency. However, it requires more computational resources and longer training time compared to LSTM models, particularly for large datasets. The self-attention mechanism, though powerful, can be resource-intensive.

10. Conclusion

Summary of Findings:

- **LSTM/GRU without Attention**: Works well for basic sequence tasks but struggles with longer sequences and dependencies.
- **LSTM with Attention**: Provides significant improvement in handling complex relationships and generates more fluent and relevant headlines.
- **Transformer**: Best overall, with significant advantages in terms of parallelism, long-range dependency handling, and fluency. However, it demands more computational resources and training time.

Practical Takeaways:

- **Best Model for Real-World Use**: The Transformer model is the best option for generating high-quality headlines, especially when performance and fluency are prioritized. However, if computational resources are limited or training time is a concern, LSTM with Attention could be a reasonable alternative.
- **Real-World Application**: For news organizations or automated content generation systems, leveraging Transformer-based models would provide the most robust results, especially with large-scale datasets.

11. Future Work

- **Using Pre-trained Models**: Fine-tuning pre-trained Transformer models (e.g., BART, T5) for domain-specific tasks can provide high-quality results with less training time.
- **Domain-Specific Data**: Fine-tuning transformers on domain-specific news data can improve headline relevance for specific news categories.
- **Improving Interpretability**: While Transformer models offer excellent performance, they can lack transparency. Research could focus on improving interpretability to better understand how they generate specific headlines.
- **Addressing Bias/Fairness**: Ensuring the models generate unbiased and fair headlines by mitigating potential sources of bias in the data or models could be an essential avenue for improvement.

12. References

- **Seq2Seq Model**:
  - Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *NeurIPS*.
  - https://arxiv.org/abs/1409.3215
- **Attention Mechanism**:
  - Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR 2015*.
  - https://arxiv.org/abs/1409.0473
- **Transformers**:
  - Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. *NeurIPS 2017*.
  - https://arxiv.org/abs/1706.03762
- **Headline Generation**:
  - Rush, A. M., Chopra, S., & Weston, J. (2015). A Neural Attention Model for Abstractive Sentence Summarization. *EMNLP 2015*.
  - https://arxiv.org/abs/1509.00685
- **Dataset**:
  - Kaggle News Headline Generation Dataset (available at Kaggle).
- **Libraries**:
  - Keras: https://keras.io/
  - TensorFlow: https://www.tensorflow.org/

.