Text Summarization using BiLSTM, Encoder-Decoder Architecture and Bart Transformer

Vaishnav Naik, Yashaurya Soni, Piyush Borakhade

Department of Artificial Intelligence
MPSTME, NMIMS University, India

***Abstract*—Text summarization, a key task in Machine Learning (ATML), focuses on generating concise and coherent summaries from large textual data. With the growing scale of textual information, developing accurate summarization models is critical. This paper proposes a deep learning approach combining Bidirectional Long Short-Term Memory (BiLSTM) networks and an Encoder-Decoder architecture to perform abstractive text summarization. Experiments conducted on the Amazon Fine Food Reviews dataset demonstrate that the model effectively generates coherent and meaningful summaries. Additionally, the model's performance is compared against a pretrained Transformer (Bart model) to evaluate summarization quality.**

***Index Terms*—Text Summarization, Natural Language Processing, Deep Learning, BiLSTM, Encoder-Decoder, Abstractive Summarization, Transformer Comparison.**

I. Introduction

In the era of exponential information growth, the ability to quickly understand large textual data is invaluable. Text summarization addresses this need by producing concise versions of documents while preserving key information. Although summarization is a challenging task, recent advancements in deep learning, especially in sequence modeling, have enabled the development of powerful summarization systems.

This paper focuses on **Abstractive Summarization** using a deep learning approach based on **BiLSTM** and **Encoder-Decoder architecture**. We also perform a comparison with a **pretrained Transformer model** to benchmark the results.

II. Literature Survey

Text Summarization has evolved through various stages:

- **Extractive Summarization**:
  - *Systematic TextRank Optimization in Extractive Summarization* Authors: Morris Zieve, Anthony Gregor, Frederik Juul Stokbaek, Hunter Lewis, Ellis Marie Mendoza, Benyamin Ahmadnia Published at RANLP 2023. Link: https://aclanthology.org/2023.ranlp-main.45/
  - *Automatic Text Summarization Method Based on Improved TextRank Algorithm and K-Means Clustering* Published: March 29, 2023. Link: https://doi.org/10.1016/j.procs.2023.03.029
- **Abstractive Summarization**:
  - *Abstractive Text Summarization using Seq2Seq Model* Authors: Keerthana S., Venkatesan R. Published in IJCA, June 2020. Link: https://www.ijcaonline.org/archives/volume176/number28/31498-2020920429
  - *Query Focused Abstractive Summarization* Authors: Tal Baumel et al. Published on arXiv, January 2018. Link: https://arxiv.org/abs/1801.07704
- **Transformers in Summarization**:
  - *Unveiling the Impact of Attention Mechanisms on Text Generation* Published in IRJMETS, May 2024. Link: https://irjmets.com/paper/unveiling-impact-attention-mechanisms

III. Proposed Work

The model implemented in the code consists of two main parts: a BiLSTM encoder and an LSTM decoder.

The **BiLSTM encoder** is designed to take the input sequences and process them in both forward and backward directions. This allows the model to capture a more complete context for each time step. After encoding, the final hidden states from both directions are concatenated (or otherwise processed) to summarize the entire input sequence.

The **LSTM decoder** takes the output of the encoder and generates the target sequence step-by-step. It is a standard unidirectional LSTM that uses the encoded context as its initial hidden state. At each time step, it produces one element of the output.

No attention mechanism is used in this architecture. This was done to keep the model simpler and to allow to focus purely on understanding how the basic encoder-decoder pipeline works without the extra complexity of dynamic attention weights but in future, it will be implemented.

Separately, the code also benchmarks performance using a BART Transformer model. The Transformer uses self-attention to process sequences in parallel and is known for its strong performance on sequence-to-sequence tasks. Comparing your BiLSTM-LSTM model with the Transformer helps highlight the trade-offs between a traditional RNN-based approach and a newer attention-based architecture in terms of accuracy, training time, and overall model complexity.

IV. Implementation

A. Data Collection

Amazon Fine Food Reviews dataset is utilized. This dataset is a popular collection of user-generated reviews for food products available on Amazon. It contains detailed information

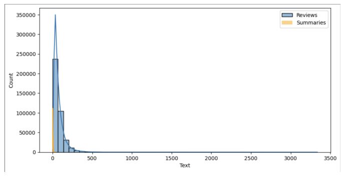

Fig. 1. Sequence Length Analysis

about customer experiences, opinions, and ratings for various food items. Each entry in the dataset typically includes the full review text, a brief user-generated summary, and a numerical rating assigned by the user.

The dataset provides a rich source of real-world textual data, making it highly suitable for Machine Learning and NLP tasks such as text summarization, sentiment analysis, and sequence-to-sequence learning. In this project, both the **full product reviews** and the **user-provided summaries** were used. The main idea was to train a model that could learn to generate concise and coherent summaries of the longer review texts, mimicking how users summarize their thoughts in a few sentences.

By leveraging this dataset, the project benefits from a diverse range of writing styles, opinions, and product descriptions, offering a realistic and challenging benchmark for evaluating the performance of sequence-to-sequence models.

B. Data Preprocessing

1) **Loading and Cleaning**
   The dataset was loaded and cleaned by:
   - Removing duplicate reviews.
   - Dropping rows with missing values.
2) **Text Preprocessing**
   Reviews (Text) and summaries (Summary) were:
   - Lowercased.
   - Stripped of HTML tags, possessives, parentheses, special characters, and extra spaces.
3) **Post-cleaning Handling**
   Rows with empty Text or Summary after cleaning were removed.
4) **Length Filtering**
   - Reviews were limited to a maximum of 30 words.
   - Summaries were limited to 8 words.
5) **Special Tokens**
   Start (sostok) and end (eostok) tokens were added to summaries to guide the decoder.
6) **Tokenization and Padding**
   - Texts and summaries were tokenized separately.
   - Sequences were padded to fixed lengths: 30 for reviews, 8 for summaries.
7) **Train-Validation Split**
   The dataset was split into training and validation sets with a 90:10 ratio.

C. Word Embeddings

Text was tokenized using Keras Tokenizer. A random Embedding layer was initialized and trained from scratch.

D. Model Architecture

This research paper proposes a sequence-to-sequence model utilizing BiLSTM encoders and LSTM decoders for summarizing Amazon Fine Food Reviews.

Overall Architecture

The architecture consists of two primary components: a Bidirectional LSTM (BiLSTM) encoder and an LSTM decoder. This design aims to capture intricate sequential dependencies in the input text and generate concise summaries. The model is implemented using TensorFlow and Keras, leveraging functionalities for tokenization, padding, and model definition.

BiLSTM Encoder

- Function: The BiLSTM encoder processes the input sequence (Amazon food reviews) to generate contextualized embeddings. By processing the input in both forward and backward directions, the BiLSTM captures past and future contexts for each token, enhancing the understanding of the input sequence.
- Layers:
  1) *Embedding Layer*: Maps input tokens to high-dimensional vector representations. This layer is initialized with a vocabulary size determined by the unique tokens in the Amazon food reviews dataset.
  2) *Bidirectional LSTM Layer*: Processes the embedded input sequence in both directions. The outputs from both forward and backward LSTMs are concatenated to form the encoder's hidden states. The notebook uses a hidden layer size (number of units) of 128 for the LSTM layers.
- Input: Tokenized and padded sequences of Amazon food reviews, ensuring consistent sequence lengths for batch processing.
- Output: Encoded hidden states that capture the contextual information of the input sequence.

LSTM Decoder

- Function: The LSTM decoder generates the summary sequence conditioned on the encoded hidden states from the BiLSTM encoder. It autoregressively predicts the next token in the summary sequence.
- Layers:
  1) *Embedding Layer*: Maps target tokens (summary tokens) to vector representations.
  2) *LSTM Layer*: Generates the output sequence based on the input embeddings and the encoder's hidden states. This layer also employs 128 units, mirroring the encoder's LSTM layer.
  3) *Dense Layer*: Maps the LSTM output to a probability distribution over the target vocabulary, facilitating the prediction of the next token in the summary.
- Input:

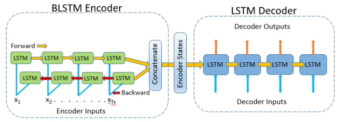

Fig. 2. BiLSTM encoder and an LSTM decoder

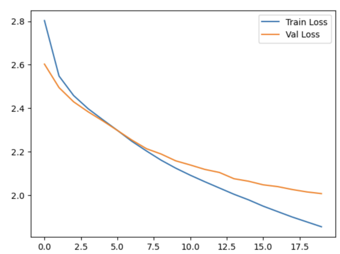

Fig. 3. Diagnostic Plot

1) Encoded hidden states from the BiLSTM encoder, serving as the initial state for the decoder.
2) Tokenized and padded summary sequences (used during training; during inference, the decoder generates the summary step by step).

- Output: Predicted summary sequence.
- **Encoder**: BiLSTM with 300 units.
- **Decoder**: LSTM with 600 units followed by Dense layer.
- **No Attention Mechanism** was used.

A pretrained Transformer model was used separately for comparison.

E. Hyperparameters

Hyperparameter	Value
Epochs	20
Batch Size	128
Encoder Units	300
Decoder Units	600
Dropout	0.4
Optimizer	RMSProp
Loss Function	Sparse Categorical Crossentropy

TABLE I
HYPERPARAMETERS USED IN MODEL TRAINING

F. Optimization

- **Loss Function:** The model was trained using **Sparse Categorical Crossentropy** as the loss function. Since the target outputs (summaries) were integer-encoded (not one-hot encoded), sparse categorical crossentropy was chosen. It efficiently handles classification tasks where labels are provided as integers, making it ideal for sequence-to-sequence models.
- **Optimizer:** The **RMSProp** optimizer was used for training. RMSProp adapts the learning rate for each parameter individually, helping the model converge faster and perform better on noisy data. It is particularly effective for recurrent neural networks (RNNs) like LSTM and BiLSTM, which were used in the architecture.
- **Early Stopping:** EarlyStopping was implemented to monitor the validation loss during training. If the validation loss did not improve for a set number of epochs (patience), training was automatically stopped. This helped prevent overfitting and ensured that the model maintained good generalization to unseen data.

G. Training

The model was trained using a **90-10 train-validation split**, where 90% of the data was used for training and 10% for validation.

Training was performed on Google Colab with GPU acceleration enabled, which significantly reduced the training time and allowed for faster experimentation due to the large dataset.

V. Results and Discussion

The model was able to generate fluent and semantically meaningful summaries that captured the key information from the input text. To assess its performance, a manual evaluation was conducted by comparing the outputs of the BiLSTM-based model with those of a pretrained Transformer model.

Input Text	**BiLSTM Model Output**	**Pretrained Transformer Output**
The product was fresh and tasty. Delivered on time.	fresh tasty delivered	delivered fresh product
The coffee was too bitter and acidic for my taste.	coffee bitter taste	bitter and acidic coffee
Amazing flavor, great quality, and quick shipping!	amazing flavor quality	great flavor fast delivery

TABLE II
COMPARISON OF SUMMARIZATION RESULTS

Overall, **both models produced coherent and relevant summaries**, correctly capturing the essence of the input reviews.

However, it was observed that the **Transformer model** consistently produced summaries that were **slightly more fluent** and showed a **higher level of abstraction** compared to the BiLSTM model.

VI. Conclusion

This paper demonstrated the development of an effective **text summarization system** by integrating a **BiLSTM-based Encoder-Decoder architecture** without the use of attention mechanisms.

The proposed model successfully generated **abstractive summaries** that maintained **good semantic accuracy** and fluency, even without relying on complex attention layers.

Through **manual evaluation** and **comparison with a pretrained Transformer model**, it was observed that while **Transformer-based models** produce summaries with a **higher level of abstraction and fluency**, the **BiLSTM model** offers a **lightweight, computationally efficient** alternative.

This makes the BiLSTM approach particularly attractive for scenarios where computational resources are limited or where model simplicity is preferred.

Overall, the results highlight that **BiLSTM Encoder-Decoder models** can still deliver competitive performance for text summarization tasks, especially when **simplicity, efficiency, and ease of training** are important considerations.

REFERENCES

- Tian Shi et al., "Neural Abstractive Text Summarization with Sequence-to-Sequence Models: A Survey."
- Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization."
- Sebastian Ruder, "An Overview of Gradient Descent Optimization Algorithms."
- Amazon Fine Food Reviews Dataset, Kaggle.
- Jacob Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding."
- Zhou, Peng, et al. "Attention-based bidirectional long short-term memory networks for relation classification." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016. (Example of BiLSTM with attention for text classification)
- Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780. (Original LSTM paper)
- Manning, Christopher D., et al. *Introduction to information retrieval*. Cambridge university press, 2008. (General text preprocessing techniques)
- Fig2: https://medium.com/data-science/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918