KANNADA ABSTRACTIVE TEXT SUMMARIZATION USING SEQUENCE TO SEQUENCE MODEL

Project Reference No.: *47S_BE_1049*

College ***:*** *B.L.D.E.A's V.P. Dr. P.G.Halakatti College of Engineering and Technology, Vijayapura*
Branch ***:*** *Department of Computer Science & Engineering*
Guide(s) ***:*** *Prof. Dakshayani. M.Ijeri*
Student(S) ***:*** *Ms. Sneha G Sajjan*
Ms. Ritika Gangavati
Ms. Aliya Bargudi

Keywords:

Deep Learning, Abstractive, Kannada text summarization, Python framework, sequence to sequence model, TensorFlow, stacked LSTM, Neural network, Attention model, encoder, decoder, regional language, NLP, preprocessing, tokenization, stemming, stop words removal.

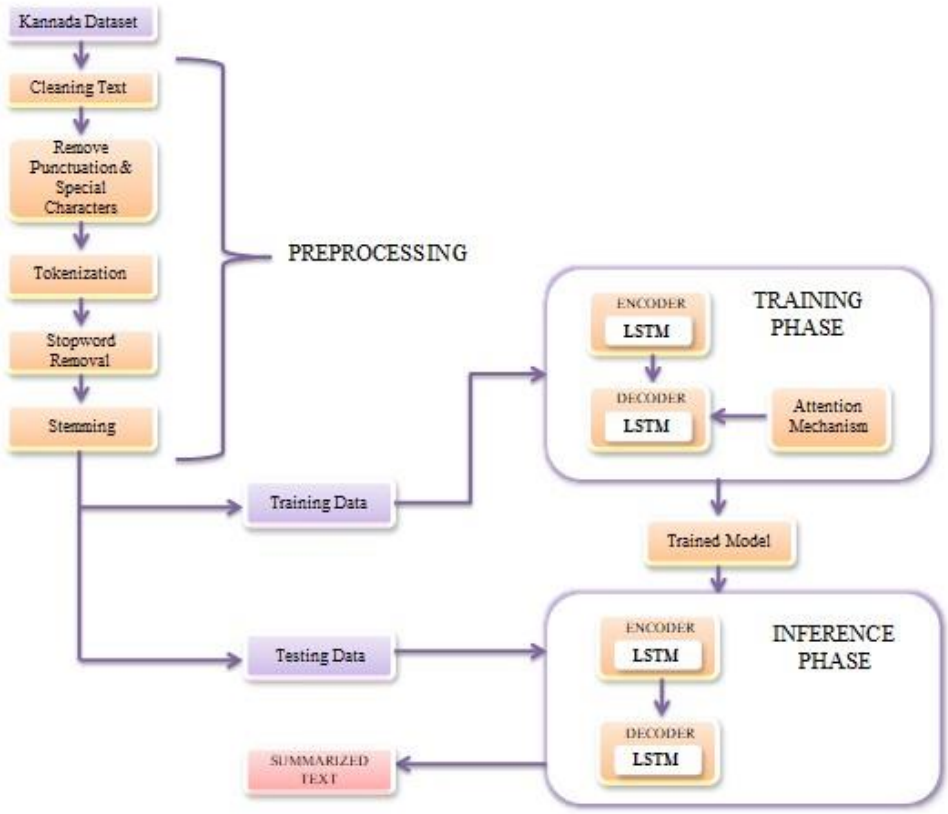
Introduction:

Kannada abstractive text summarization aims to generate concise and coherent summaries of Kannada texts using advanced natural language processing techniques. This project implements a sequence-to-sequence (Seq2Seq) model with an attention mechanism, designed specifically for the Kannada language. The Seq2Seq model comprises an encoder and a decoder, both built with LSTM layers, to capture the contextual information of the input text and generate meaningful summaries. The attention mechanism allows the model to focus on different parts of the input sequence dynamically, improving the quality of the summaries. We preprocess the text data, including cleaning, tokenization, and stemming, to ensure it is suitable for model training. The model is trained on a dataset of Kannada texts and their corresponding summaries, using early stopping to prevent overfitting. Post-training, the model is capable of generating abstractive summaries for new Kannada texts, providing a powerful tool for information condensation in the Kannada language. This project showcases the application of deep learning techniques to a low-resource language, contributing to the broader field of natural language processing and text summarization.

Objectives:

1. To develop a text summarization model for Kannada language using abstractive method.
2. To generate Kannada summary including numeric value.

Methodology:



The methodology depicts a text summarization system. It breaks down the process into two main stages: training and testing/inference.

In the training stage, text data in Kannada is collected from a dataset. This data goes through a preprocessing stage where it is cleaned. This cleaning process involves removing punctuation and special characters. The text is then tokenized, which breaks it down into individual words or phrases. Next, stop words, which are common words like “the” or “and” that don't carry much meaning, are removed. Finally, stemming is applied to reduce words to their base form.

After preprocessing, the text is ready for training. An encoder, which is a type of recurrent neural network (RNN) called a long short-term memory (LSTM) network, processes the text data. An attention mechanism is also used, which helps the model focus on important parts of the text. This encoded data is then used to train a decoder LSTM, another recurrent neural network.

Once the model is trained, it can be used to generate summaries of new text data. This new, unseen text data goes through the same preprocessing steps as the training data. The

preprocessed text is then fed into the encoder LSTM, which encodes the text data. The decoder LSTM then uses the encoded text data to generate a summary of the text.

Overall, the methodology outlines a system that can be used to automatically generate summaries of text data.

Results and Conclusions:

ROUGE Scores:

- ROUGE-1: F1 Score = 0.639
- ROUGE-2: F1 Score = 0.599
- Average ROUGE F1 Score = 0.615

The ROUGE F1 Score is 0.615, which suggests that the summaries generated by the model are fairly similar to the human-written references.

In the realm of data cleaning and preprocessing, this approach integrates crucial steps such as stemming and the optional removal of stopwords, which are vital for effective text processing. By analyzing the distribution of text lengths, it ensures that preprocessing decisions are well-informed. The model architecture employs an encoder-decoder structure with attention mechanisms, making it particularly suitable for abstractive summarization tasks. The incorporation of LSTMs and dropout layers aids in capturing long-term dependencies and mitigating the risk of overfitting. Additionally, beam search decoding is utilized to enhance the quality of the generated summaries by evaluating multiple candidate sequences, thereby improving the overall output.

What is the innovation in the project?

- Tailoring the sequence-to-sequence model to Kannada, a language with unique linguistic features, presents a novel challenge in natural language processing.
- Implementing attention mechanisms tailored to Kannada text to enable the model to focus on relevant parts of the input text during the summarization process, thereby improving the quality of generated summaries.

Scope for future work:

- Model can be trained better to preprocess the stressed words(ottakshara) to generate precise summary.
- Model can be better trained to summarize multi lingual numeric data.
- Hyperparameter Tuning:
  Experimenting with different hyperparameter values (latent dimension, embedding dimension, number of LSTM layers) can potentially improve model performance.
- Advanced Attention Mechanisms:

Exploring advanced attention mechanisms like self-attention or hierarchical attention could further enhance the model's ability to capture complex relationships within the text.