Neural Abstractive Text Summarizer for Telugu Language

Mohan Bharath B¹, Aravindh Gowtham B², Akhil M³

¹ M. Tech, IIIT Allahabad, Allahabad, U.P, India
² B. Tech, Shiv Nadar University, Greater Noida, U.P, India
³ M. Tech, University of Hyderabad, Hyderabad, Telangana, India

¹ bharathhbm@gmail.com, ² bg739@smu.edu.in, ³ akhil9351@gmail.com

Abstract. Abstractive Text Summarization is the process of constructing semantically relevant shorter sentences which captures the essence of the overall meaning of the source text. It is actually difficult and very time consuming for humans to summarize manually large documents of text. Much of work in abstractive text summarization is being done in English and almost no significant work has been reported in Telugu abstractive text summarization. So, we would like to propose an abstractive text summarization approach for Telugu language using Deep learning. In this paper we are proposing an abstractive text summarization Deep learning model for Telugu language. The proposed architecture is based on encoder-decoder sequential models with attention mechanism. We have applied this model on manually created dataset to generate a one sentence summary of the source text and have got good results measured qualitatively.

Keywords: Deep Learning, LSTM, Telugu, Neural Networks, NLP, Summarization

1 Introduction

Textual data is ever increasing in the current Internet age. We need some process to condense the text and simultaneously preserving the meaning of the source text. Text summarization is creating a short, accurate and semantically relevant summary of a given text. It would help in easy and fast retrieval of information. Text summarization can be classified into two categories.

- **Extractive Text Summarization** methods form summaries by copying from the parts of the source text by taking some measure of importance on the words of the source text and then joining those sentences together to form a summary of the source text.

- **Abstractive text summarization** methods create new semantically relevant phrases, it can also form summaries by rephrasing or by using the words that were not in the source text. Abstractive methods are actually
harder. For an accurate and semantically relevant summaries, the model is expected to comprehend the meaning of the text and then try to express that understanding using the relevant words and phrases. So, abstractive models can have capabilities like generalization, paraphrasing.

Significant work is being focused on extractive text summarization methods and especially with English as the source language. There is no reported work for Telugu abstractive Text summarization using Deep learning models and also there are no available datasets for Telugu text summarization. Our goal is to build a model such that when given the telugu news article it should output semantically relevant sentence as the summary/title sentence for the corresponding telugu article. We have proposed a Deep learning model using encoder-decoder architecture and we have achieved good results measured qualitatively.

We have manually created the Dataset because of the fact there are no available datasets. Training Dataset has been created from the Telugu News websites by taking the headline as the summary and the main content as the source text and we have created a dataset with 2000 telugu news articles with their corresponding summaries which are taken as the headline of the respective article. We have created the dataset in such a way that the articles belonging to the different domains i.e, Politics, Entertainment, Sports, Business, National are more or less equally distributed to maintain a balance to the dataset.

To create word embeddings for the telugu words, we have made use of word-embeddings by FastText, which has created word embeddings for nearly 157 languages with each word-embedding of 300 dimensions.[1]

2 Related Work

As our Work is based on Abstractive Text Summarization using Deep learning models on Telugu Language on which there is no reported work. Our Deep Learning Models are mainly inspired by these three papers, Rush et al.[2] used an Encoder-Decoder Neural Attention Model to perform Abstractive Text Summarization on English text-data and found that it performed very well and beat the previous non Deep Learning-based approaches, Konstantin Lopyrev proposed a Encoder-Decoder recurrent neural network with attention mechanism to Generate Headlines for English News[3],and attention mechanism itself is inspired from Bahdanau seminal paper[4].
3 Approach

In this section we provide a brief overview of the Model/Architecture used and its individual components.

3.1 Recurrent Neural Network Encoder-Decoder

3.1.1 RNNs

Text is a sequential type of data. Recurrent Neural Networks are a type of Neural Networks used for handling sequential data. RNNs can take a variable length input sequence $X = (x_1, x_2, x_3, \ldots, x_t)$ and can output sequence $Y = (y_1, y_2, y_3, \ldots, y_t)$. It uses an internal hidden state ($h$) to capture both the current input and the previous hidden state. A simple mathematical representation of recurrent network can be seen below.

$$h_t = \sigma_h (W_h x_t + U_h h_{t-1} + b_h)$$ (hidden state at timestep t)

$$y_t = \sigma_y (W_y h_t + b_y)$$ (output at time-step t)

$W_h$ is the weight matrix connecting input layer to hidden states and $U_h$ is the weight matrix connecting the hidden states of current time step to the hidden states of previous timestep. $b_h$ and $b_y$ are biases of hidden states and of the output layer. $W$, $U$ and $b$ are parameters and they are the same at each time step. $\sigma_h$ and $\sigma_y$ are activation functions. RNNs can also be multi layered they are called Deep RNNs, these are layered RNNs, where each layer extracts information from the previous layer.

3.1.2 LSTMs

Long short-term memory (LSTM) is a type of RNN architecture with complicated hidden unit computations. So, by introducing gates which are input, forget, output and memory cells, it allows memorizing and forgetting over a long distance of training and the model can effectively handle the vanishing gradient problem. LSTM is a basic unit of our Encoder-Decoder model to perform summarization.
The motivation behind using an LSTM is that it captures long-term dependency pretty well and the information in the starting of the sequence is able to traverse down the line. This is done by being selective and restricting the information flow in the LSTM unit. There are three gates in an LSTM.

**Forget Gate Layer:**

- \( f_t = \sigma(W_f[h_{t-1}; x_t] + b_f) \)
- \( C_t = C_{t-1} \odot f_t \)

\( C_t \): Cell State at time step-\( t \)
\( W_f \): weight matrix
\( h_{t-1} \): hidden state of previous timestep.
\( x_t \): input at timestep-\( t \)
\( f_t \): forget-gate
\( \odot \): Dot-product.
\( \sigma \): Sigmoid activation

**Input Gate Layer:**

- \( i_t = \sigma(W_i[h_{t-1}; x_t] + b_i) \)
- \( C_t = \tanh(W_i[h_{t-1}; x_t] + b_i) \)
- \( C_t = C_{t-1} \odot i_t \)

\( i_t \): input gate.
\( C_t \): Cell state at timestep \( t \).
\( W_i \): weight matrix between previous state and current input.

**Output Gate Layer:**

- \( o_t = \sigma(W_o[h_{t-1}; x_t] + b_o) \)
- \( h_t = o_t \odot \tanh(C_t) \)

3.1.3 **Encoder-Decoder Model:**
Encoder-Decoder model is based on neural networks which aims at handling the mapping between the highly structured input and output. In the vanilla encoder-decoder model, the encoder RNN first reads the source text word by word and then encodes the information in a hidden state and passes the information forward. Decoder starts from the final hidden state of the encoder and at every timestep it computes a probability distribution on the total words in the vocabulary by taking a SoftMax function which gives probability values to all the words in the vocabulary and the most probable word is selected for that timestep and this continues until the end of sentence token is selected by the decoder or until the no of timesteps reaches the threshold. All the words generated so far will form the summary sentence!

- \( x = (x_1, x_2, \ldots, x_T) \) be a length \( T \) input sequence to the encoder network.
- \( y = (y_1, y_2, \ldots, y_U) \) be a length \( U \) output sequence the decoder network generates.
- Each Encoded representation \( h_T \) contains information about the input sequence with focus on the \( T^{th} \) input of the sequence.
- \( h = (h_1, h_2, \ldots, h_T) \) be the encoder network output of length \( T \).

Fig. 2. Encoder-Decoder Architecture

In the Encoder-Decoder framework the encoder tries to summarize the entire input in a fixed dimension vector \( h \), Decoder takes as input the output of the previous step and the hidden-state vector from the previous time step. So, at every timestep of the Decoder, the word selected in the previous time step and the hidden-state vector of the previous time step is given as input in the current timestep.

### 3.1.4 Attention Mechanism [4]

The basic Encoder-Decoder model performs well on very-short sentences but it fails to generalize well for longer sentences/paragraphs.
The only input to the Decoder at the first timestep is a fixed size vector from the Encoder stage last timestep.

- This fixed size vector may not be able to capture all the relevant information of the longer source text.
- At each step of the decoder only certain parts of the input are relevant to generate an appropriate word.
- How do we make the model learn what to focus on at each step of the decoder? 
  **Attention!**

Attention model calculates the importance of each input encoding for the current step of the Decoder by doing a kind of similarity check between decoder output at this timestep and all the input encodings. This similarity check is done by taking Dot product between the current hidden state of the decoder and all the input encodings. Doing this for all of the input encodings and normalizing, we get an importance Vector. We then convert it to probabilities by passing through SoftMax, which would give probability distribution. Then we form a context vector by multiplying with the encodings.

- \( \text{importance}_{it} = V^* \tanh(e_i W_1 + h W_2 + b_{\text{attn}}) \)
- Attention Distribution \( a_{it} = \text{SoftMax}(\text{importance}_{it}) \)
- Context vector \( h^*_t = \sum e_i a_{it} \)

Context Vector is then fed into two layers to generate distribution over the vocabulary from which we sample.

- For the loss at time step t, \( \text{loss}_t = -\log P(w_t^*) \), where \( w_t^* \) is the target summary word. i.e. negative log probability of the target summary word.

- Loss at all the time steps is summed up.

- Now, we can do backpropagation and get all the required gradients and to minimize the loss we can apply gradient descent and learn the parameters of the network.

**4. Training**

Language: Python
Libraries: TensorFlow, Keras, etc.

- Dataset which is created manually, contains 2000 Telugu news articles and their corresponding summaries which are taken from the Headline of the respective articles.
- Dataset is divided into two parts: 1700 for Training and 300 for testing.
- Model has 128 hidden units.
- Loss function is cross-entropy.
- Training is done for 40 epochs.
- Training of Encoder-Decoder Architecture is done end to end i.e, loss is propagated to the encoder as well.

5. Evaluation

Evaluation of the results i.e, generated summaries of the source text have been done qualitatively. Here, a sample of the generated summaries has been given below.

Table 1. Results

| Source Text | Original Summary | Generated Summary |
|-------------|-----------------|------------------|
| టిక్టాక్ యాప్ను భారతీయులు అధికంగా వినియోగిస్తున్నారు. రోజురోజుగా టిక్టాక్ యాప్ను ఉపయోగిస్తున్నా వారి సంఖ్య పెరుగుతంది. యూజరుు తమ వీడియోల్లో ప్పపంచంల్ల అతయ ంత పెదిదైన్ సోలార్ టెలిసోో ప్పసదిిడని డేనియల్ కే ఇనౌయే సోలార్ టెలిసోో (డీకేఐఎస్టీ) మందెన్ా డూ చూడని స్తరుయ డి అద్భుతం ఫొటోలు! | భారతీయులు టిక్టాక్ వీడియోలను చూస్తుంది. కొంతమందికి తమ ప్పపంచంల్లనే అతయ ంత పెదిదైన్ సోలార్ టెలిసోో ప్పసదిిడని డేనియల్ కే ఇనౌయే సోలార్ టెలిసోో ప్పసదిిడని డీకేఐఎస్టీ (డీకేఐఎస్టీ డీ) | భారతీయులు టిక్టాక్ వీడియోలు డాన్స్ లు, పాటలు, చాలంజ్లు చేస్తున్నారు. సోషల్ మీడియాల్లో పాపులర్ అయిన్ విషయం తెలిసందే. సాధారంగా ఈ టిక్టాక్ వీడియోలను చూస్తుంది. కొంతమందికి తమ ప్పపంచంల్లనే అతయ ంత పెదిదైన్ సోలార్ టెలిసోో ప్పసదిిడని డేనియల్ కే ఇనౌయే సోలార్ టెలిసోో ప్పసదిిడని డీకేఐఎస్టీ (డీకేఐఎస్టీ డీ) మందెన్ా డూ చూడని స్తరుయ డి అద్భుతం ఫొటోలు! | भारतीय टिकटॉक वीडियोलाई देखती हैं। कुछ लोगों की इस संख्या में बढ़ती है। यूजर्स अपने वीडियो में कई फोटो देखते हैं। साधारणतः टिकटॉक वीडियोलों को देखने पर वीरल बनते हैं और शेयर करने पर। कुछ लोग इनके बारे में रोचक सामील हैं। काल की जनों में चार्टर और विशेषता के लिए इसका इस्तेमाल किया जाता है।
అధి అవిష్కరణకి కారణం దీనిద్వారా ఉపరితలానికి సంబంధించిన అరుదైనింటి ఫొటోలను చూసే అవకాశం మానవాళికి దక్షిణందు అంశాలను సంబంధించిన అంచన వేసే అవకాశం ఉంటుందని ఆస్తసోటన్నటున్నా ప్పస్తుంది. ఈ భారీ టెలిస్ట్రానికి సమీపంగా చూస్తుదుంటాను. అంతరాగతి కిందిందు అవకాశం ఉండతుంది అంతరాగతి అంచనా ఉండగా తెలిపారు. ఒలింపికి ఏడాది నేపథ్యింలో భారత బ్యాడింటన్నా సింగిల్స్ కోచ్ ఆఫ్ఇండ్యా (సయ్) గతంలో క్రీడా మింత్ర క్రీడాశాఖను కోరించింది. దానిని సపించించిన మింత్ర క్రీడాశాఖ సింటోస్ ను నియమిస్తుంది. అతడు ఒలింపికస్ముగిసే వరకు కోచ్గా సేవలు అందించాడు. సింటోస్ మారిండో వారింలో భారత బ్యాడింటన్నా జట్టానికి కలుసు డు. అతడు పరయవేక్షణలో పించియన్ని పీవీసింధుతో పాట్టిండి. ఇతర సింగిల్స్ షట్టాల కూడా టోకోయసింధించిన మింత్ర క్రీడాశాఖ సింటోస్ పరయవేక్షనుతుంది. అతడు పరయవేక్షనలో పించియన్ని పీవీసింధుతో పాట్టిండి. సింటోస్ శిక్షణతో సింతృపిందితే అతడునుంటున్న బయల్ వెళ్లదింతో ఏరపడ్ని కోచ్ ఖాళీని భర్తిచేయాలించాలింటూ సోపర్ సాధారణాఫ్ఇండ్యా గతంలో క్రీడా మింత్ర క్రీడాశాఖను కోరించింది. దానిని స్పించించిన మింత్ర క్రీడాశాఖ సింటోస్ ను నియమిస్తుంది. అతడు ఒలింపికస్ముగిసే వరకు కోచ్ గా సేవలు అందించాడు. సింటోస్ మారిండో వారింలో భారత బ్యాడింటన్నా జట్టానికి కలుసు డు. అతడు పరయవేక్షనలో పించియన్ని పీవీసింధుతో పాట్టిండి.
అనుమాన్నస్పద్ కారయాసపాలను కొనసగించి మొబైల్ అపుల కషన్ ల గురించి వాటిచపట్టపట్టం అధునాతన ఫీచర్ను పేటీఎం పేమించింది. రోగపై ఈ ఫీచర్ ను అందిసో ింది. మోస్పూరిత లావాదేవీల్ను పసగటి.. ఏయాప్ దాిరా స్మాచరిం చేశార్ల డీజిల్ ట్రీకు స్మానింగా ఇది పనిచేసుకోవచ్చి కింపనీ స్హ. వయవసప్పకుడు సదా ధరారాజన్ మీడియాకు తెలపార్ల.

2019లో వివరించర్లు ఉండి రూ.150 వద్ద అమితామిత అని వివరించర్లు. ఉదాహరణకు భారత్ లో సెలెసియల్ ఈ ఏడాది దేశీయ రియల్టీ రింగింగ్ ఆసించినింత వృది ధనినించలేదు. వినాయవిగింతగా లక్షల ల పర్యాటకాల్తో దేశీయ రియల్టీ రింగింగ్ లో వృది అవకాశాల్ను నీర్లగారాయని అనరాక్ పారం కనసలె ింటస్చేయింది. దేశిదిని ఏడు పించవాయప్పు ింగా ఆరికండి మనింది టిరకరకాల్ కారణాల్తో దేశీయ రియల్టీ రింగింగ్ లఋ కవల్పు 20,48,300 గృహాలు. 2019సంవత్సరం 2019 సంవత్సరం వినాయుగా మాయా.
We have observed that the generated summaries are more or less semantically relevant to the source text in many cases in the test data, despite the limited training data our model did a good job.

6 Conclusions

We have implemented abstractive text summarization for Telugu language using encoder-decoder architecture with attention mechanism. Specifically, there are no datasets for Telugu language that have paired human-generated summaries. We have got semantically relevant good results on most of the test data measured qualitatively, although there are very few generated summaries which were not at all relevant. Given the fact the Dataset has been created from the scratch and the limitation of the dataset size, we have got good results. For text summarization in Telugu language, one of the difficulties is the lack of quality summaries of large given dataset, in future we would like to create a very large dataset corpus for this purpose comparable to the standard English text summarization datasets available, we would also work towards a numeric metric specifically to telugu text summarization tasks which would also capture the semantic relevance of the summary to the given source text. In future we shall explore the possibility of using the transformer architecture for telugu text summarization task.
We have taken permission from competent authorities to use the images/data as given in the paper. In case of any dispute in the future, we shall be wholly responsible.

References

1. Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, Tomas Mikolov. “Learning Word Vectors for 157 Languages.” arXiv:1802.06893v2 [cs.CL]
2. Alexander M. Rush, Sumit Chopra, Jason Weston. “A Neural Attention Model for Abstractive Sentence Summarization.” arXiv:1509.00685v2 [cs.CL]
3. Konstantin Lopyrev. “Generating News Headlines with Recurrent Neural Networks” arXiv:1512.01712v1[cs.CL]
4. Dzmitry Bahdanau, Kyuhyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473v7 [cs.CL]