Report on

"Generation of Abstract Summaries for Food Reviews using a Sequence to Sequence Architecture with Local Attention Mechanism "

Submitted in partial fulfillment of the requirements for ***Sem VII***

Topics in Deep Learning

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Abhishek Narayanan	**01FB16ECS016**
Abhishek Prasad	**01FB16ECS017**
Abijna Rao	**01FB16ECS019**

Under the guidance of

Srikanth H.R.
Assistant Professor
PES University, Bengaluru

January – May 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PROBLEM STATEMENT :

This project is aimed at implementing a deep learning model for performing Abstractive Summarization of fine food reviews from Amazon using a Sequence to Sequence architecture, incorporating local attention mechanism for an improved performance compared to the standard vanilla version of the encoder decoder architecture. We also implement a bidirectional LSTM for the encoder to account for context in both forward and backward directions, while the decoder uses the standard unidirectional LSTM.

INTRODUCTION :

With the exponential growth in technology and rampant usage of the internet each day, textual data is growing rapidly with each passing day. Therefore there is an immediate need for algorithms to compress or condense text data while retaining the semantics and important information. Text summarization refers to the automatic generation of summaries in natural language from an input document while preserving its semantics.

There are two prominent types of summarization algorithms.

• Extractive text summarization algorithms are capable of *extracting* key sentences from a text without modifying any word.A large number of researchers have been focussing on extractive approaches due to the ease of defining hard-coded rules to select important sentences than generate natural language, which is a complex task. Apart from that, extractive approaches output grammatically correct and coherent summaries. But due to their restrictive nature, they fail to summarize long and complex texts well.

• Abstractive summarization, instead, involves a complex process understanding the language, the context and generating new sentences. This frees the model of the constraint of using pre-written text but involves using large-scale data during training.

In recent literatures, Neural Sequence to Sequence architectures have been shown to be promising in Abstractive Text Summarization. But they are plagued by problems of often generating repetitive and absurd summaries, often grammatically incorrect. In an attempt to tackle this challenge of abstractive text summarization, we explore and implement from scratch, a sequence to sequence architecture with attention (local attention in specific). We also implement a bidirectional LSTM for the encoder to account for context in both forward and backward directions, while the decoder uses the standard unidirectional LSTM.

PROPOSED METHODOLOGY :

The following subsections elaborately portray the methodology incorporated in our experimentation for performing abstractive summarization.

DATASET USED :

For the purpose of training a sequence to sequence model to learn summarization of text reviews, we use the Amazon Fine Food Reviews data, procured from Kaggle website. This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all approximately 500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

DATA PRE-PROCESSING :

As a preliminary preprocessing step performed to cleanse the data of unwanted noise components, we first remove all unnecessary non-printable characters which are irrelevant to our task. We also filter the dataset further by removing reviews which are shorter than 25 characters or longer than 300 characters. Similarly, data instances having summaries longer than 15 characters have also been removed to retain only abstract short summaries.

Neural architectures like other algorithms are incapable of processing raw text directly. Therefore we need to generate numerical representations of text (reviews and their summaries) which are capable of preserving the semantics effectively in vector space.

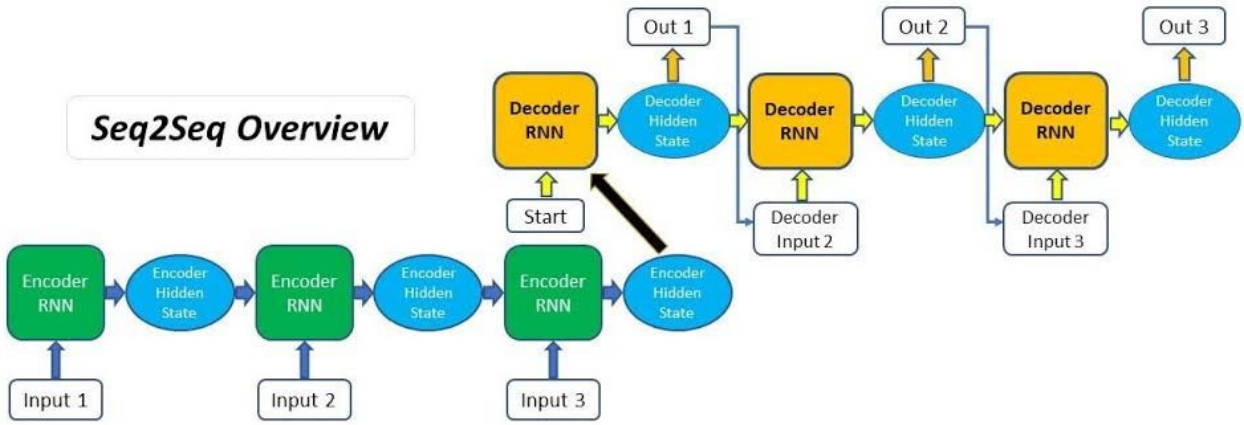
For this purpose, we use the Global Vectors (GloVe) for word representation. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

However, for our experiments, we do not train a GloVe model on our corpus and instead use a pre-trained model trained on a corpus of 6 Billion words, which provides us word vectors of 100 dimensions as semantic representations. Since the vocabulary on which the model is pre-trained on is significantly high, we assume most of the model would fit for most of the words in our corpus. The rare words in our corpus whose embeddings are unavailable in the pre-trained model are neglected.

We find the review and summary with maximum length and pad the shorter reviews with a <PAD> tag. Words which are not in the vocabulary are assigned an <UNK> tag signifying “unknown”. Another tag referring to the end of sentence or <EOS> is also added to the vocabulary. All of these are assigned embeddings from a random distribution and these vectors are maintained consistent throughout to represent these tags.

MODEL ARCHITECTURE :

The standard sequence to sequence Encoder-Decoder Architecture has been portrayed diagramatically in the figure below :



The encoder RNN cells are implemented as Long Short Term Memory cells (LSTMs) rather than the vanilla simple RNNs which are prone to the vanishing gradient problem and thus are incapable of capturing context over long sequences of text.

We employ Bi-Directional LSTM encoder in order to capture context in both forward and backward directions. This essentially is implemented by feeding the input in forward direction first through an LSTM layer and then is reverse order. The hidden states thus obtained are then concatenated and stacked in a tensorArray for later use, when attention would be applied.

The final hidden state, also termed as the thought vector is considered to be a semantic representation of the whole sentence and is fed to the decoder to start the decoding process for generating abstract summaries. The first input is fed as a start symbol (SOS : Start of sentence) to indicate to the decoder that it has to start generating words. The decoder continues to generate words until an <EOS> or end of sentence tag is generated.

It is to be noted that the input at each time step Xi is a word vector, in our case a GloVe embedding.

Training is performed batch-wise in batches of size 32 data instances. We use the NAdam optimizer, which is essentially a combination of Nesterov Momentum and RMSProp for controlling learning rate decay and adding momentum. We apply Dropout layers with dropout rate of 0.3, L2 Regularization and Early Stopping as steps to prevent overfitting of the model. Training is forcefully terminated when the validation accuracy does not improve over 5 continuous epochs. The loss function used is Cross Entropy which has been proven to be

effective when using the softmax activation in the output layer. Due to the high training time involved, the model is trained for 10 epochs for 10 hours on the Google Research Collaboratory environment with 25GB RAM and with GPU backend enabled.
It is also to be noted that the decoder does not output words as text as it is, but a probability distribution as a vector generated by softmax with dimension as the vocabulary size. The index of the word in the vocabulary corresponding to the position with maximum probability is used to retrieve the actual word generated.

Inspired by existing research papers, which have shown significant improvement in abstractive summarization using attention with sequence to sequence architectures, we additionally incorporate attention in our model in an attempt to generate better results.
Inspired by the research by Luong et al., we incorporate local attention in our architecture which has been elaborated further.

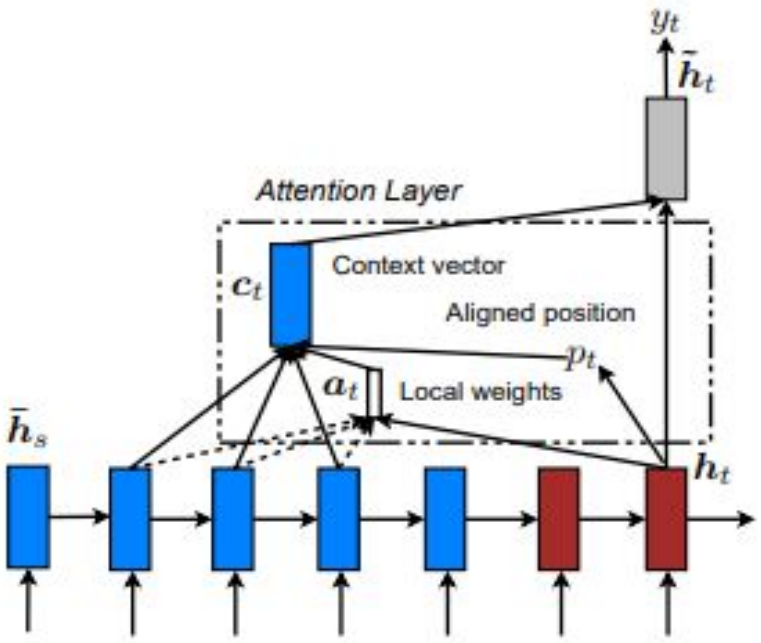


Figure : Local attention model – the model first predicts a single aligned position pt for the current target word. A window centered around the source position pt is then used to compute a context vector ct , a weighted average of the source hidden states in the window. The weights at are inferred from the current target state ht and those source states in the window.

The global attention has a drawback that it has to attend to all words on the source side for each target word, which is expensive and can potentially render it impractical to translate longer sequences, e.g., paragraphs or documents. To address this deficiency, we propose a local attentional mechanism that chooses to focus only on a small subset of the source positions per target word.

The local attention mechanism proposed by the authors of the aforementioned research selectively focuses on a small window of context and is differentiable. This approach has an advantage of avoiding the expensive computation incurred in the soft attention and at the same time, is easier to train than the hard attention approach. In concrete details, the model first generates an aligned position pt for each target word at time t.

The context vector ct is then derived as a weighted average over the set of source hidden states within the window [pt−D, pt+D]; D is empirically selected.

We implement the Predictive alignment (local-p) technique proposed in the research– the model predicts an aligned position as follows:

$$p_t = S \cdot \text{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W_p} \boldsymbol{h}_t)),$$

Wp and vp are the model parameters which will be learned to predict positions. S is the source sentence length. As a result of sigmoid, pt ∈ [0, S]. To favor alignment points near pt , we place a Gaussian distribution centered around pt . Specifically, our alignment weights are now defined as:

$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right)$$

The standard deviation is empirically set as σ= D/ 2 . Note that pt is a real number; whereas s is an integer within the window centered at pt . The value of D has been chosen to be 5 in our experiments.

RESULTS :
The improvement in results obtained after each epoch have been summarized below.

Sample Text

```
for those who like a bold , strong flavored coffee . this <UNK> is a must
try . for those who are new to k cups try a variety pack first tp help you
better decide your favorite coffee blends .
```

Sample Predicted Summary

```
great coffee
```

Sample Actual Summary

```
awesome !
```

Sample Text

i was happy with the product . it is surprisingly difficult to find this mint green tea by it self . so i love that i now have 6 boxes , my kids are even drinking it ! ! the price was very reasonable as well . will buy again ! !

Sample Predicted Summary

great !

Sample Actual Summary

very pleased

Sample Text

good stuff . i wish they would run a special on it though . i liked it . it is all i can find . locally i can not find any cola syrup . but it tastes pretty good to me . i always wait until the price is right then i grab a few . i enjoy drinking it .

Sample Predicted Summary

great syrup

Sample Actual Summary

diet cola syrup

Epoch: 7

Average Training Loss: 0.746
Average Training Accuracy: 42.50
Average Validation Loss: 0.750
Average Validation Accuracy: 43.58

Sample Text

gloria jeans has a great product of hazelnut flavored coffee . the product is liked by all my colleagues . the transaction via amazon was prompt and efficient . will buy again from amazon .

Sample Predicted Summary

great coffee

Sample Actual Summary

great coffee

Sample Text

i start my morning with this coffee each day . great taste and great price . i have ordered this coffee several times and will buy again .

Sample Predicted Summary

great

Sample Actual Summary

great coffee

Sample Text

it was so easy to grow and my cat loved it . i highly recommend this purchase . your cat will love you forever !

Sample Predicted Summary

cat

Sample Actual Summary

cat grass

Sample Text

this coffee is the best . i first tasted it at the tyler back and spine hospital in tyler , texas . if you truly like coffee you wo n't be disappointed . it has a wonderful flavor and no bitterness .

Sample Predicted Summary

great coffee

Sample Actual Summary

great coffee !

Sample Text

i ordered lipton green tea for my mother ; it 's her favorite tea . the price was very good , and the large amount of tea bags will last her a long time .

Sample Predicted Summary

very tea

Sample Actual Summary

great tea !

Sample Text

it is a good drink just a little too <UNK> < br / > i use lots of ice and a little water so it cuts down bit ...

Sample Predicted Summary

nice stuff

Sample Actual Summary

good stuff

Average Training Loss: 0.723
Average Training Accuracy: 44.02
Average Validation Loss: 0.738
Average Validation Accuracy: 44.13

CONCLUSION AND FUTURE WORK :

In this project, we have implemented from scratch using tensorflow, a sequence to sequence model with local attention mechanism. We obtain an accuracy score of about 44.13% on a test set. However, it is observed that the model outputs reasonably good summaries. The difference between the output obtained and the score could possibly be due to the fact that word to word positional matching is not a suitable metric for this task, because the model is significantly penalized when a word generated at a certain time step is different from the expected one, even though the generated output is in a different form but conveys similar meaning, which should still be considered as a viable summary. Rather, the evaluation metric should be based on a similarity score between the generated and actual output. For instance, the BLEU score is a common metric in neural machine translation rather than simple accuracy due to the aforementioned reason.

Therefore as a future work in order to extend this project, we intend to explore relevant evaluation scores and loss functions. We also intend to explore other complex architectures and attention mechanisms in related works, which could result in a performance comparable to the existing state-of-art in this problem domain.