Abstractive Text summarization

Udit Gupta
M.tech CDS
14454
udit2008.gupta@gmail.com

Ramabhadra V
M.tech CDS
15168
ramavyasa@gmail.com

Abstract

The current metrics used to compare text summaries only take into account the overlapping words from the reference summaries. In abstractive summarization this may not capture the essence of the generated summary. We explore 3 different deep models for text summarization and present results. We analyze the 'quality' of the generated summaries from our best model using a different evaluation metric which takes into account the sentiment.

Introduction

Text summarization can be broadly classified into 2 categories; Extractive and Abstractive. Extractive summarization creates a condensed version of the input text by only using words from the source text to create the summary. This has the drawback that it is too restrictive since it just selects and rearranges source text sentences to generate summaries. On the other hand, Abstractive summarization is not limited to words from the input but instead generates a summary based on semantic understanding of the source text. This has the advantage that it can paraphrase, compress, and generalize the source text and hence the generated summaries are more 'human-like'. Abstractive text summarization can be thought of as a two-step process: a sequence of text is first encoded into some kind of internal representation. This internal representation is then used to guide the decoding process back into the summary sequence.

Given instances of source text and their summaries, we separate the dataset into 60-20-20 train-validation-test set. We solve the summarization problem as a supervised learning problem where the source text is considered as a feature vector and the summaries are the target. For this problem we use Recurrent Neural Networks because they exhibit dynamic temporal behavior for a time sequence. This is because RNNs can use their internal state to process sequences of inputs. However, vanilla RNNs are not good at handling long term dependencies due to vanishing gradient problem. Therefore, for this project we use LSTMs, a special kind of RNN, which are capable of learning long-term dependencies.

Related Work

A hierarchical framework was proposed by Cheng et al.[1] which makes use of CNN to generate sentence representation and RNN to represent the document.
The approach proposed by Cao et al [2] is based on Recursive Neural Networks for ranking sentences for multi-document summarization.
Rush et al [3] utilized Neural attention model with a contextual input encoder to generate abstractive summaries.

Dataset

In our project, we use Amazon food review data set. Dataset consists of reviews of foods from amazon. There are more than 500,000 reviews which are generated in a span of 10 years till 2012. We are using instances where, text length is between 15 and 40 words and summary length is 3 to 7 words. This will result in 40k samples. The vocabulary consists of 30k words.
We are using 3 features from the dataset; Actual text, actual summary and consumer rating.

Proposed Architectures

We started our experiments with simple adaptation of Machine translation model [4] since Encoder-Decoder recurrent neural network architecture developed for machine translation has

proven effective when applied to the problem of text summarization. We first implemented the seq2seq model which encodes the content of an article (encoder input) and one word (decoder input) from the summarized text to predict the next word in the summarized text. This model was first implemented without using pre-trained word embeddings and then another model was implemented using Glove pre-trained word embeddings which are fine tuned for our model.

Sequence to sequence model adapted from machine translation model

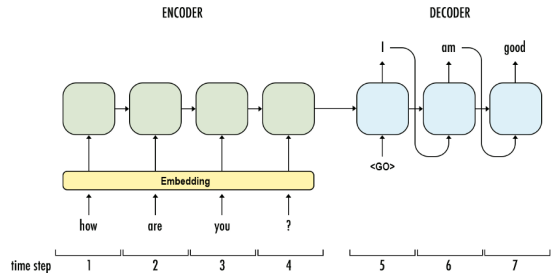

Figure 1: Illustration of a simple seq2seq model.

The encoder encodes the input sequence to an internal representation called 'context vector' which is used by the decoder to generate the output sequence. In this model a RNN layer acts as an encoder. It processes the input sequence and returns its own internal state. We discard the outputs of the encoder RNN and only recover the state. This state serves as the context or conditioning of the decoder in the next step. Another RNN layer acts as decoder. It is trained to predict the next words of the target sequence, given previous words of the target sequence. Specifically, it is trained to turn the target sequences into the same sequences but offset by one timestep in the future, a training process called "teacher forcing". The decoder uses the state vectors as initial state from the encoder. Therefore, the decoder learns to generate targets[t+1...] given targets[...t], conditioned on the input sequence.

Text Summarization Encoder-Decoder architecture

One shot

The one-shot RNN is a very simple encoder-decoder recurrent network model which encodes the source text and decodes the entire content of the summarized text. The decoder uses the context vector alone to generate the output sequence. This model puts a heavy burden on the decoder. The decoder will not have sufficient context for generating a coherent output sequence as it must choose the words and their order.The model architecture is shown in Figure 2.

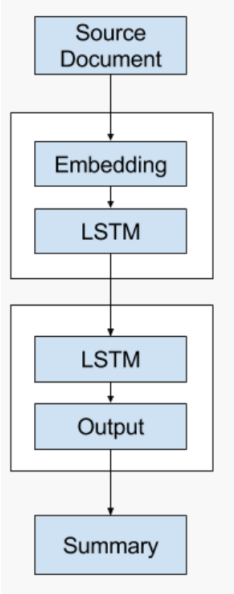

Figure 2: Architecture of one shot RNN.

Recursive Model

In this model, encoder generates a context vector representation of the source document which is fed to the decoder at each step of the generated output sequence. This allows the decoder to build up the same internal state as was used to generate the words in the output sequence. This ensures that the decoder is primed to generate the next word in the sequence. This process is repeated by calling the model again and again for each word in the output sequence until maximum length(7 in our case) or 'END' token is generated. The model architecture is shown in Figure 3

Experiments

All the models mentioned are trained for 50 epochs on the CLSERV which took 2hr for each model. We evaluated our models using Recall Oriented Understudy for Gisting Evaluation (ROUGE) metric. We used F1 scores for ROUGE-1 (This is the number of unigrams common for hypothesis and reference summary), ROUGE-2

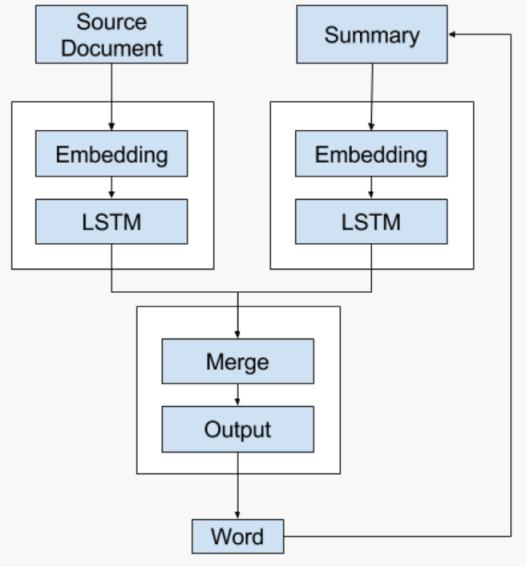

Figure 3: Architecture of Recursive encoder-decoder model.

(This is the number of bigrams common for hypothesis and reference summary), ROUGE-l (This is the length of longest common subsequence between hypothesis and reference summary). All the scores were generated using `rouge` package in python.

Results

We present accuracy on train and test sets for comparison of different models for the task of text summarization. We use the ROUGE metric for evaluation of the generated summaries as shown in

Accuracy

The graphs represent the train and test accuracies.

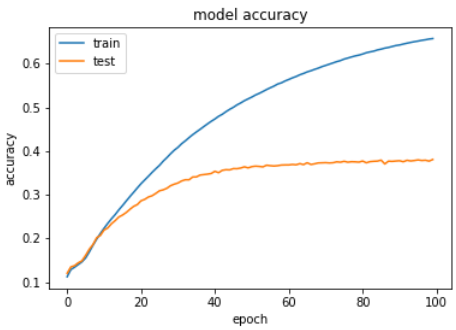

Figure 4: Accuracy vs epochs for Seq2Seq without glove embeddings

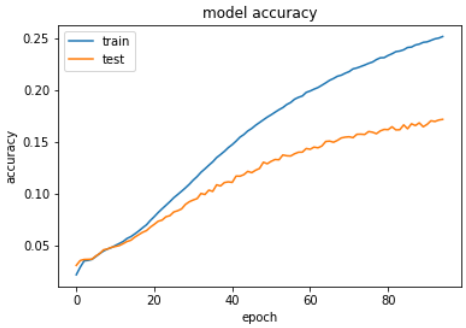

Figure 5: Accuracy vs epochs for with glove embedding

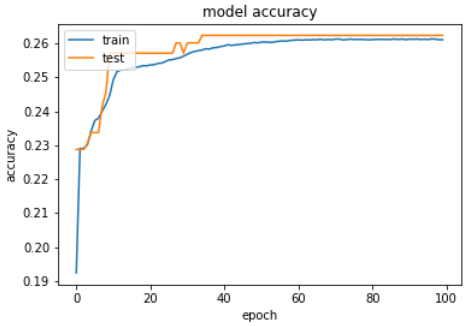

Figure 6: Accuracy vs epochs for 1 shot RNN

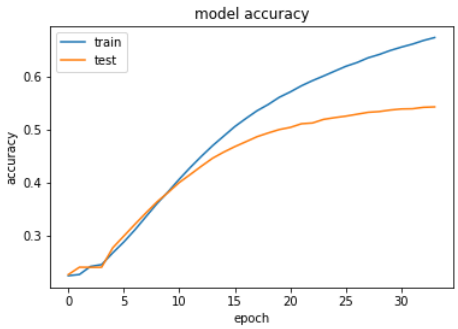

Figure 7: Accuracy vs epochs for Recursive RNN

Rouge Scores

Comparison of Rouge F1 scores

Model	R1	R2	RL
Seq2Seq (W/O Glove)	31.2	18.5	29.4
Seq2Seq (With Glove)	17.4	9.1	15.3
Recursive	38.6	29.2	37.9

Comparison of summaries

1. **Actual summary:**absolutely the most delicious coffee.
   Generated Summary:best tasting french vanilla coffee.

2. **Actual summary:**cannot get enough of it.
   Generated Summary:good product and good price.

3. **Actual summary:**was not what i expected.
   Generated Summary:do not like berry flavor.

Transfer Task

Current generated summary evaluation metric is ROUGE. Even though, it is a good metric for comparing news summaries, We believe that it is not an efficient evaluation metric for review summary evaluation. Hence, We present a novel evaluation metric by briefly elaborating transfer task of summaries generated.

Abstractive summaries generated can be used for sentiment analysis. Sentiment of the text might sometimes be difficult to evaluate if the input is lengthy. The most important trait of good summary of reviews is that, they should essentially capture the sentiment of the input.

This feature is absolutely essential while generating summaries for reviews. This is because, when the input is "The coffee in this shop tastes like piss" and the given summary is "Coffee is bad", a summary which says "coffee is extremely awful" will very strongly capture the essence of the input. However, this is not reflected in ROUGE-1 score. ROUGE-1 score is just 0.5 and ROUGE-2 score is 0.33. Hence, we have come up with a new metric which uses sentiment scores to compare.

We are using ratings provided by the user while writing the review to evaluate the summaries generated. As mentioned, summary of a review should essentially capture the sentiment of the input. Hence, we are using mean squared error of sentiment of summary generated and user provided rating to evaluate generated summaries. We are generating sentiment scores using this resource.[5]

$$MSE(y) = \frac{\sum_{i=1}^{n} (y_i - r_i)^2}{n}$$

where,
y_i is corresponding i_{th} sample
r_i is the rating of i_{th} sample

We sampled 200 samples randomly from from full dataset and calculated MSE(Actual input), MSE(Actual summary) and MSE(generated summary). We present the results in the table below.

MSE Comparisons for 200 instances

	MSE
Actual Text	1.74
Actual Summary	2.21
Generated Summary	1.67

Future work

We have implemented 2 types of encoder-decoder models. However, we have not considered state-of-the-art Attention mechanisms. One improvement upon our models is to incorporate them.
We also observed that increasing the vocabulary size makes the model take more time for each epoch and they also perform worse. One could extend the models discussed to be robust to vocabulary size.

References

1. Cheng, J., and Lapata, M. 2016. Neural summarization by extracting sentences and words. 54th Annual Meeting of the Association for Computational Linguistics
2. Cao Z, Wei F, Dong L, Li S, Zhou M (2015) Ranking with recursive neural networks and its application to multi-document summarization. In AAAI conference on artificial intelligence
3. Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv 1509.00685
4. Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv 1409.0473.
5. http://text-processing.com/docs/sentiment.html