Abstractive Summarization of Amazon Fine Food Reviews

Munot Rushab Preetam
14405
rushab@iitk.ac.in

R Animesh
14511
animeshr@iitk.ac.in

Shibhansh Dohare
14644
sdohare@iitk.ac.in

1 Introduction

Summarization in general is wide topic and has numerous applications. From a general perspective - text summarization, document summarization, sentiment/emotion analysis, keyword extraction, headline generation for news articles, etc all come under the topic of summarization. And summarization itself can be viewed as an instance of transforming one sequence into another. Here we focus on a specific example for summarization which is generating short (5-6 words) summaries for food reviews from the 'Amazon Fine Food Reviews' database(Amazon Fine Food Reviews, datase from Kaggle,).

2 Task Description:

The dataset 'Amazon Fine Food Reviews' from Kaggle(Amazon Fine Food Reviews, datase from Kaggle,) contains food reviews on Amazon until October '12. The fields include Id, ProductId, UserId, ProfileName, Helpfulness Numerator (number of users who found the review helpful), Helpfulness Denominator (number of users who indicated whether they found the review helpful), Score (between 1 and 5), Time, Summary and Text. The dataset contains multiple parameters of which the following seem useful:

- Text of the reviews - Input
- Summary - Output
- Score - Output

The problem can be informally stated as follows: Given text of food review, generate a short summary of the review and predict a score for the review between 1 and 5.

Formally: Given a sequence of sentences for a food review generate a sequence of output words summarizing the review and predict its score.

Thus given a sequence of sentences $s_1, s_2, \ldots s_n$ predict the output sequence of words $w_1, w_2, \ldots, w_m$ where n and m are not fixed. Each s_i is further a collection of words as in $s_i = \{s_{i_1}, s_{i_2}, \ldots s_{i_k}\}$.

3 Models and Techniques proposed

The challenge here is that the output sequence length is not fixed for a particular length of the input sequence. Thus for the same length of input, the ouput length may vary. To fix this we plan to use either of the following two models: Seq2seq model (Sutskever et al., 2014) or attention based encoder-decoder RNN model(Chopra, S. Mozer, M.C., 2016).

Recurrent neural networks are difficult to use due to the limitation posed by fixed size input/output. The above models use LSTMs (seq2seq) or attention based neural networks(Chopra, S. Mozer, M.C., 2016). LSTMs *learn* when to read, write or forget keys stored at the nodes. Similarly, attention based RNNs, have a pointer that decides when to write something to the output and when not to.

The other part is to vectorize sentences and words for training. The word to vector model (Mikolov et al., 2013) seems appropriate for this purpose. The model gives a linear relationship between words and exploiting the similarity between the becomes feasible and easy.

4 Platform and Libraries:

We shall use python for implementation. As for the libraries the following libraries will be used though more libraries may be used later: Numpy, Scipy, Scikit-learn, Gensim, nltk, Theano.

5 Acknowledments

We would like to thank Prof. Harish Karnick for giving us the opportunity to work on this project. We would also like to thank our TA, Mr. Pawan Kumar to help us in finalizing the project.

References

Ilya Sutskever, Oriol Vinyals, Quoc V. Le. 2014. *Sequence to Sequence Learning with Neural Networks*,

Amazon Fine Food Reiews. Kaggle *https://www.kaggle.com/snap/amazon-fine-food-reviews.*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. *Distributed Representations of Words and Phrases and their Compositionality.*

Chopra, S. and Mozer, M.C. 2016. *Abstractive Sentence Summarization with Attentive Neural Networks, NAACL 2016*