UNIVERSITY OF SCIENCE AND TECHNOLOGY AT ZEWAIL CITY

COMMUNICATIONS AND INFORMATION ENGINEERING

CIE 555
Neural Network and Deep Learning

Abstractive Text Summarizer

Team Information	*ID*
Fatma Moanes NourEl-Din	201700346
Mohamed Abd El-Rahman	201700030
Mahmoud Ashraf Hamed	201700989
Hassan Youssef	201601850

Course Instructor

Dr. Mohamed Elshenawy

Contents

1 Introduction

Automatic Text Summarization is one of the most challenging topics in NLP. Abstractive Text Summarization is the task of creating summaries that contain new sentences than those already found in the original text, so the model needs to understand the language itself in order to create meaningful summaries without copying sentences from the original text.

2 Our Plan

2.1 Initial Plan

Our initial plan was to use the CNN-Dailymail dataset which contains news articles and a summary for each article. We were going to fine tune the state-of-the-art Pegasus model, proposed by google researchers, on our dataset. We also wanted to fine tune the same dataset using different transfomer based models. Besides, building a Sequence-to-Sequence model and using the dataset to train the model. So, our goal was to compare and interpret the results obtained from different models.

2.2 Challenges

We faced many problems with our initial plan. Firstly, the Pegasus model is such a huge model that just fine tuning it requires a massive 68 GB of RAM. However, we were using the free version Google Colab for training, and such amounts of RAM are not available for us. Moreover, another problem was the dataset itself. The problem is that each training example is a very large news article, so for most models to be able to train on this dataset they require a lot of computation power as well as a lot of memory, both are not available to us using the free Google Colab.

2.3 New Plan:

We had to change and adapt our plan to the limitations that we had. Therefore, we decided to change the dataset and use the Amazon Fine Food Reviews dataset. This dataset consists of reviews made by customers on food products, and for each review the customer also writes a very short concise description of his review. Hence, we can use this dataset for learning text summarization by using the reviews as texts, and the short description as the summary of the review. Furthermore, we decided to use sequence-to-sequence based models, because they contain less trainable parameters than transformer based models so they needed less computation power.

3 Text Pre-processing

Our data pre-processing steps can be split into two main steps: data cleaning and data tokenization to prepare the data for the model.

3.1 Data Cleaning

The following steps were used to clean the data:

1. Normalize all characters to lowercase to allow the model to learn the same representation for same words with different cases.
2. Expand contractions to allow the model to learn the same representation for the words with and without contractions.
3. Remove the possessive ('s).
4. Remove any text between the parentheses and the parentheses themselves, because text between parentheses is usually out of context and might confuse the model.
5. Remove punctuation and special characters, because they don't carry much information that can improve the model performance.
6. Remove stopwords, because stopwords do not contribute to the meaning of the sentence.
7. Adding start and end tokens to the summaries.

After the process of data cleaning comes the process of data tokenization. Accordingly, we tokenized the data by assigning a unique integer to each different word, but we excluded any words that occurred less than 4 times in the whole dataset from the tokenization process. After tokenizing the data, we needed to fix a constant length for all reviews and a constant length for all summaries, as required by our model. After inspecting the distribution of sentence lengths, we found that 60% of reviews have lengths below 30 words long, and 94% of the summaries have lengths below 8 words. According to these numbers, we decided to fix all reviews to only 30 words, and all summaries to only 8 words. This was achieved by clipping or zero padding sentences to ensure equal lengths.

4 Model

We use a sequence-to-sequence based model. Our model consists of an encoder decoder architecture with an attention layer. Figure 1, shows the architecture of our model.

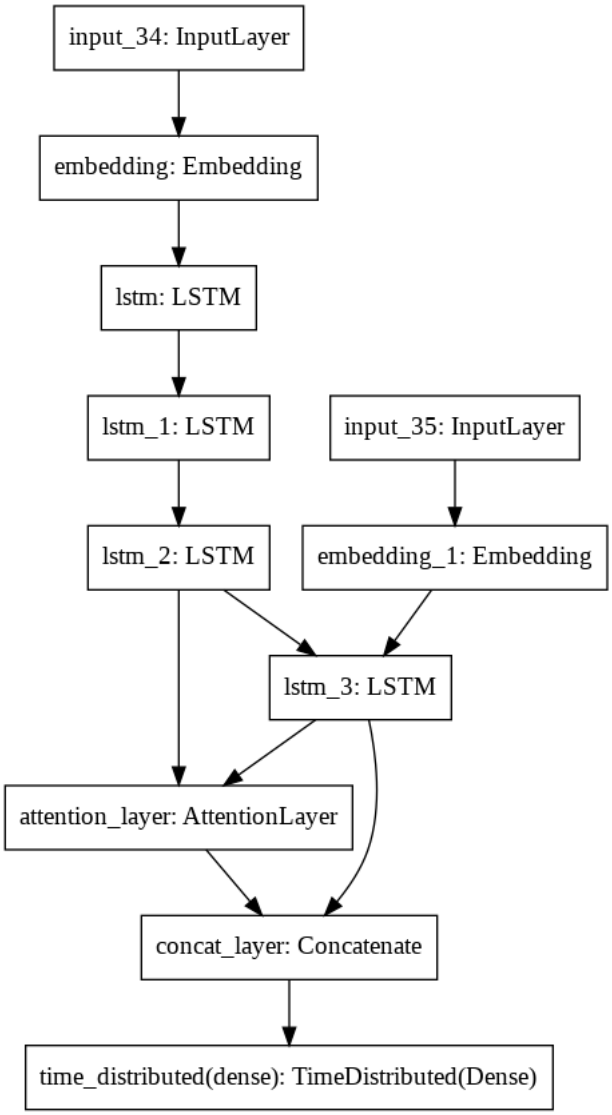

Figure 1: Base model

The encoder consists of an embedding layer followed by three LSTM layers. The embedding layer attempts to learn a 100-dimensional word embedding representation that allows the model to understand the relation between different words in a proper way. Then, the output of the embedding layer is fed into three consecutive LSTM layers that learn to encode the word embedding of the whole sentence, and output a representation for the sentence than can be easily processed by the decoder.

The decoder consists of an embedding layer followed by an LSTM layer and an attention layer. The embedding layer has the same job as the encoder's embedding layer, but it learns word embeddings for the summaries instead of the reviews. Moreover, the LSTM layer in the decoder processes the output of the embedding layer and the hidden state of the last LSTM layer in the encoder. The attention layer helps the decoder to focus its attention on words in the review that are relevant to the current word in the summary being predicted instead of focusing on all the words each time the decoder is predicting a new word. Finally, the output is fed into a dense layer whose activation is softmax. The purpose of this layer is to output the probability of each word in the vocabulary being the next word in the summary.

While training the model, the embedding layer of the decoder takes as input the target summary to predict. However, during inference the model works by taking the text to be summarized as input, and feeding the whole text to the encoder. A start token is fed into the decoder input and the LSTM layer of the decoder is initialized by the output of the last LSTM layer in the encoder. Further, the model will output the probability of each word being the first word in the summary. Consequently, the word with the largest probability is selected and fed as input again to the decoder which uses this input to predict the second word in the same manner. This operation terminates when either the end of summary token is generated or the maximum summary length is achieved.

5 Results

To assess the results quantitatively we used the ROUGE-1 metric. ROUGE-1 is suitable because most summaries are around two to three words long, so using ROUGE-2 will not be suitable. Also, qualitative analysis is carried out by visually inspecting the produced summaries.

5.1 Base Model Performance

The base model performance achieves a ROUGE-1 score of 11.6%. Below a sample of the model output is shown.

Review: tasty however per nutrition info package one serving cookie calories half cookie fat half cookie grams eat whole thing calories grams fat
Original summary: tasty but
Predicted summary: not gluten free

The base model does not produce good results or good summaries, and it appears from the predicted summary that it does not really represent the review.

5.2 Adding More Layers to The Encoder

To improve the results we added two more LSTM layers to the encoder, and the ROUGE-1 score slightly increased to 12.2%. Below a sample of the model output is shown.

Review: tasty however per nutrition info package one serving cookie calories half cookie fat half cookie grams eat whole thing calories grams fat
Original summary: tasty but
Predicted summary: good snack

In the above predicted summary, we can see that now the model is able to predict a more meaningful summary that represents the review. We also note that the model predicted summary contains the word snack which is another way of saying the word cookie, so the word embedding layer was capable of learning that a cookie is a snack.

5.3 Using 10 dimensions for word embedding

We tried to test if increasing the word embedding dimension will significantly worsen the results, given that it will significantly increase the training time. The results of the ROUGE-1 score is 12.3% which is quite similar to the results obtained before, however if we look at the following examples we can see that the model is now producing summaries that are less meaningful.

Review: tasty however per nutrition info package one serving cookie calories half cookie fat half cookie grams eat whole thing calories grams fat
Original summary: tasty but
Predicted summary: popchips

5.4 Training For 50 Epochs

In the previous trials we were training the model with an early stopping scheme that stops the scheme if the validation loss decreases for a few epochs to prevent the model from overfitting the training data as well as reduce the time of training. In this trial, however, we decided to train the model for 50 epochs without early stopping to investigate the effect on the model output. The model achieved a ROUGE-1 score of 16.6%, which is greater than all the trials we did before. Below a sample of the model output is shown.

Review: tasty however per nutrition info package one serving cookie calories half cookie fat half cookie grams eat whole thing calories grams fat
Original summary: tasty but
Predicted summary: nice crackers

Review: love bars reason get stars price tried save money buying instead store turns difference price nearly cheaper much would think anyway taste quality great
Original summary: love them
Predicted summary: not bad at all

From the above predicted summaries we can see that, because the model had more epochs for training it was able to learn a much better word embedding allowing it to learn that cracker and cookie are the same thing for example.

5.5 Training Without Removing Stopwords

Training the model on a dataset without any stopwords will cause the model summaries to be free of stopwords as well. We decided to investigate if leaving the stopwords affect the results or not. The model achieved a ROUGE-1 score of 15%. Below a sample of the model output is shown.

Review: my family have lot of food allergies and nana cookies are always good snack these are by far the best it taste like brownie very yummy even my friends and family that do not have love it have no idea why someone gave it such poor review guess they are able to eat snickers bar and calorie wise it is not that bad heck of lot better then other cookies would totally recommend these my month old loves them and so do and my husband stars
Original summary: love love love these cookies
Predicted summary: great for gluten free

We can see that the above predicted summary does contain a stopword. However by inspecting many examples, we believe that the stopwords in the predicted summaries do not add much meaning to them since the summaries are already very small.

6 Workload Distribution

Hassan Youssef was the member responsible for choosing an architecture to be implemented based on the limitations we had.
Fatma Moanes was the member responsible for the data preprocessing.
Mahmoud Ashraf was the member responsible for implementing the model.
Mohamed Elshaffei was the member responsible for tuning and testing the model.