E-Commercial Review Summarization

Author: Yuan Dang

Instructor: Edwin Lo

May, 2023

Table of Contents

Abstract

This project focuses on summarizing product reviews with transformer-based encoder decoder models; These models are designed to use on for similar tasks such as documents summarization or translation, and they are used on formal written language, but reviews are more casual. The project uses transfer learning techniques to modify existing models and extend them to downstream task.

1 Introduction

While online shopping is becoming increasingly popular, the massive amount of user-generated reviews can cause information overload, which makes it difficult for users to find relevant information amidst an abundance of words. Therefore, there is a great need of review summarization, which can condense and extract the most relevant and informative content from a large collection of reviews. For sellers, they can get the actual needs of customers from the reviews information to improve their product quality and therefore enhance their competitiveness. For customers, they can draw on other people's products experience to better assist them in making purchase decisions.

2 Data Collection

2.1 Source

This dataset posted by Jianmo Ni from University of California at San Diego. (https://nijianmo.github.io/amazon).

2.2 Overview

The original data files contains products information and reviews from Amazon released in 2018 and is split into json files by the following 29 product types:

Amazon Fashion; All Beauty; Appliances; Arts, Crafts and Sewing; Automotive; Books; CDs and Vinyl; Cell Phones and Accessories; Clothing, Shoes and Jewelry; Digital Music; Electronics; Gift Cards; Grocery and Gourmet Food; Home and Kitchen; Industrial and Scientific; Kindle Store; Luxury Beauty; Magazine Subscriptions; Movies and TV; Musical Instruments; Office Products; Patio, Lawn and Garden; Pet Supplies; Prime Pantry; Software; Sports and Outdoors; Tools and Home Improvement; Toys and Games; Video Games

The original dataset has 233,100,000 samples, since it is too large to train, we select 10000 samples from each product type randomly to form a dataset with 290,000 samples. We split the dataset into train, validation, and test set by ratio 6:2:2.

2.3 Sample from Dataset


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "vote": 5,
  "style": {
    "Format:": "Hardcover"
  },
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Figure 2-I Sample From Original Files

3 Data Preprocessing

3.1 Drop Missing Value

reviewText	101
summary	77
overall	0
type	0
asin	0

Table 1 Missing Values in Columns

3.2 NLP Text Cleaning

- To Lower Case:

  Convert all alphabet characters to lower case.

- Remove Parenthesis
- Fix contractions:

  Using Python package *contractions* to split contractions(e.g. I'm, it's in to I am, it is).

- Transform Emoticons:

  Some text contains emoticons such as ';)', ':(', import dictionary from *emot* package to decode them.

- URL

  Some text contains URL and they are not well formatted, use python Regular Expression to extract and remove them.

- Consecutive Characters

  Some text contains typo or repeated characters(e.g. gooood), use Regular Expression to convert them into normal format.

  ```
  ' you can get a bunch of diffferent opinions'
  ```

- Stop Words

We keep stop words to keep sentence coherence since it is important in text summarization

- Non-relevant Text

There are some meaningless reviews. Remove them from dataset.

	overall	reviewText	summary	type	asin	text
5505	5.0	***	Good deal	Prime_Pantry	B000RA6L42	

Figure 3-I Non-relevant Text in Dataset

3.3 Overview of Data

reviewText	summary
I have a 50lb bully that loves to jump from the front seat to t...	I have a 50lb bully that loves to jump from ...
works great.	Five Stars
Fits the dog just fine and unlike it so it's not showing him like...	Fits the dog just fine and unlike it so it's not showi...
Water polisher it is, let me tell you people, this thing made a ...	Works great....
My German Shepherd starts licking his lips as I approach him ...	Sparkeling White Smile

Table 2 Extracted Reviews From Dataset

4 Visualization

4.1 Dataset Distribution

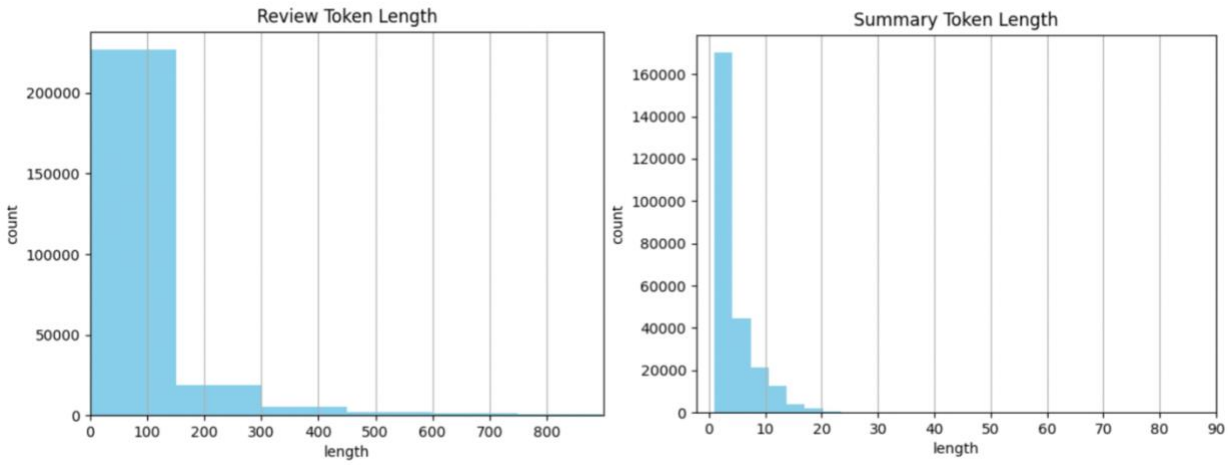

Figure 4-I Text Length Distribution

95% of reviews have less than 500 words, there are some outliers with length more than 700; most summaries have 1-25 words. Those values help to determine the input shape of encoder and decoder. In this case, we choose encoder input length 512, and decoder input length 64.

4.2 Word Cloud

We could use WordCloud to take a better look of content of reviews. In word cloud frequency of word is considered as its importance, and the importance of each word is shown with font size.

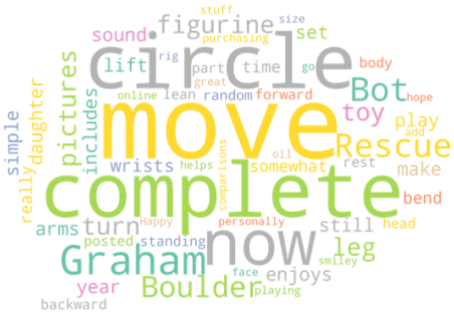

Figure 4-II A Sample of Encoder Tokens

Figure 4-III A Sample of Decoder Tokens

5 Tokenization

In this project we use two different tokenizer the embedding the review text.

5.1 T5 Tokenizer

Instead of using traditional word embedding method like Word2Vec or GloVe, we use T5 with a byte-level encoding technique that breaks down text into subword units using byte-pair encoding (BPE). BPE is a compression algorithm that recursively merges the most frequently occurring pair of bytes in a given text corpus, creating a vocabulary of subword units that are used to represent the original words. This approach allows the T5 tokenizer to handle rare and out-of-vocabulary words, and to generate meaningful representations of text even in cases where the vocabulary is limited.

5.2 Bert Tokenizer

The BERT model uses position embeddings and segment embeddings to generate contextualized embeddings for each subword unit. Position embeddings capture the position of each subword unit in the input sequence. The contextualized embeddings generated by the BERT model are highly effective at capturing the meaning of text in context by considering the surrounding subword units when generating each embedding.

6 Model

After trying to design our custom MLP, CNN or LSTM models from scratch, their performances are so poor. I implement transfer learning with pre-trained models, in other words, transformers. Because transformers have already been trained on large scale of text, they work well than simple deep neural networks like MLP, CNN. However, just using transformers is not enough, we need to modify them for our summarization task. Therefore, we use transformers as models' body, and add some special heads: MLP, CNN. To avoid overfitting, we also need to add dropout layers right after the transformers body and between head layer. According to Hendrycks and Gimpel (2016), GELU activation function is mathematically better than RELU activation function in nonlinearity task, which we will use in this summarization task.

6.1 Transformer

Transformers are capable of capturing word dependencies and contextual information since they use self-attention mechanisms that enables them to better understand the semantic meaning of the text and generate more accurate summaries. For both heads, we use different pre-trained transformer as our base: T5-small and Bert. T5-small has 60 million parameters compares to Bert model which has 110 million parameters. Model with larger number of parameters are computationally expensive(takes more time and memory), in this case, Bert will take about 11 hours while T5-small take about 6 hours, but Bert will generate higher quality output.

With each head, we tried both T5-small base and Bert base transformer on single review summarization and collection of reviews summarization. The collection of reviews contains all reviews under a single product. The model structures are shown as below.

6.2 Transformer + MLP

The MLP head is a fully-connected neural network that is added on top of the transformer base. Its purpose is to take the output of the transformer and transform it into the desired output format, in this case to control the output length. The general data flows as:

1. Input sequence: *" I love this product. Sturdy, spacious with lots of compartments and also lined with magazine holders."*
2. Transformer output: A sequence of hidden states that represent the meaning of the input sequence.
3. MLP head input: The final hidden state of the transformer.
4. MLP head output: A sequence of words that represent the summary of the input sequence: *"Sturdy"*

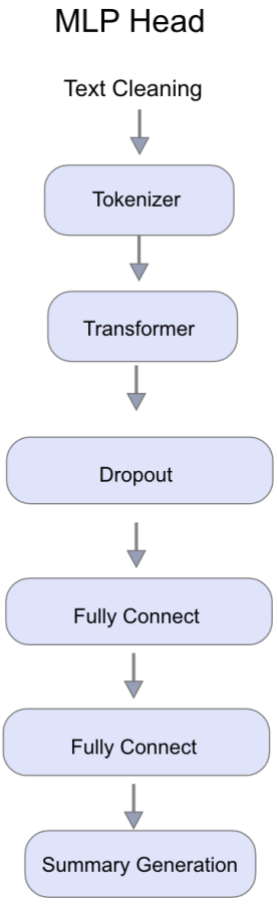

Figure 6-I Transformer with MLP Head Model Structure

6.3 Transformer + CNN

The model structure and data flow in this case is very similar to transformer + MLP head. The difference is that since MLP head consists of fully connected layers, it is good at capturing global relationships between the input and output, and CNN head contains convolutional layers that are good at identifying patterns that occur at different positions in the input.

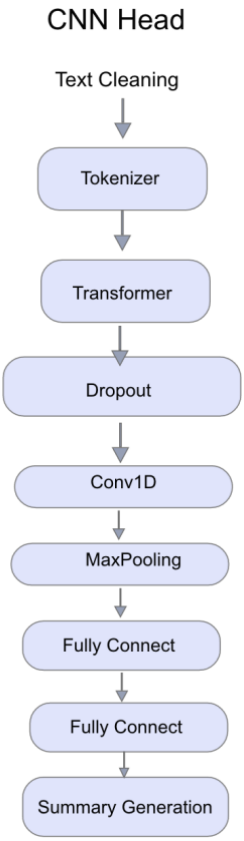

Figure 6-II Transformer with CNN Head Model Structure

6.4 Generation Algorithm

Also called decoder algorithm, the default generation algorithm is greedy search, in order to improve the quality of generated summary by the models, we can try some different methods.

6.4.1 Beam Search

Beam search works by keeping track of a given number of the most promising candidate sequences, called the "beam," and exploring each of these sequences by generating the next

likely word given the previous words in the sequence. The algorithm then selects the most promising sequences from the beam to continue exploring, and repeats this process until a stopping criterion is met (such as reaching a maximum length or achieving a certain level of confidence in the generated sequence).

6.4.2 Top-k Sampling & Top-p Sampling

Top-k sampling considers only the k most likely words at each step, and selects one of them randomly with probabilities proportional to model probabilities. Top-p sampling is similar to top-k sampling, but instead of selecting the top k words, it selects from the smallest set of words whose cumulative probability mass exceeds a pre-defined threshold, typically denoted as p. Since top-k sampling can sometimes choose rare and wired word as next word, we use a mixed top-k and top-p together as a generation method.

6.4.3 Comparison

- **Original Review**:

I am really happy with this mount. The mount held each phone very well, and I believe that it has enough movement to hold something as big as a Note.

The only issue I have had with the mount is that it left a mark on the dash of my car. I had the mount placed on the dash for approximately a month, and it stayed perfectly the whole time, but when I went to reposition it I noticed there was a ring mark left on the dash. It has now been another month and the ring has finally disappeared. I tried a number of different cleaners on the ring and none of them seemed to help. I am not sure if this an issue with the mount, or if the dash of my car is weird, but because of this I have since moved the mount to the windshield.

Despite my issue, I would recommend this mount to anyone looking for something universal.

- **Beam Search:** I'm really happy with this mount
- **Mixed Sampling:** Good mount, but it keeps the ring on its dash

- **Original Review**:

  This is some of my favorite deodorant I have ever used. It is just so smooth and silky. I never get white stains with this, and gel deodorant just does not seem to work as well. The smell is amazing too.

- **Beam Search:** Smooth and silky
- **Mixed Sampling:** This is my favorite deodorant I have ever used.

The result shows that while both algorithm is able to generate meaningful summaries, the sampling method is capable to catch more details, such as the complaint about mark in first example. Therefore, we will use mixed sampling method when generating output with our models.

6.5 Evaluation

Use ROUGE Metrics to compare model-generated summary with human-written summary provided in original dataset and measure how well they overlaps.

- ROUGE 1: overlap between single words
- ROUGE 2: measure the overlap between pairs of consecutive
- ROUGE L: measure number of Longest common subsequence

  Each of those ROUGE metrics also contains

- Precision: How well summary is focused on capture only important information
- Recall: How well summary is focused on capture only important information
- F-measure: Overall balance between precision and recall

6.5.1 Single Review Summarization

For single review summarization, let's focus on f-measure which measure the overall balanced score.

f-measure	ROUGE 1	ROUGE 2	ROUGE L
T5-small	11.519	5.364	10.726
T5-small + MLP	12.638	6.556	12.015
T5-small + CNN	31.071	23.091	30.077
BERT	6.449	1.037	5.133
BERT + MLP	28.659	20.854	27.930
BERT + CNN	30.545	22.611	30.239

The really low score of Bert baseline is because Bert pre-trained model generate very long summaries, so the precision score is much lower than recall score which leads to low f-measure score which is overall balance between precision and recall.

We can see that after add task-specific heads on original models, the scores are highly improved. And MLP head has lower score than CNN head generally, which might because MLP head is better-suited for tasks where the goal is to predict a single label or category for an input text, while a CNN head is better-suited for tasks where local features are important and need to be extracted from the input text. So we will focus only on the CNN header this time.

On the other hand, We can see T5 based model has a little bit higher score than BERT based model, which means we get a better performance

6.5.2 Collection of Reviews Summarization

As previously mentioned, We also generated summaries for collection of reviews where each collection contains all reviews under a single product. With same model structure, in this case we use recall score, because we want to focus on how well the model capture all the important information. For a single review, we want the summary be as short as possible, so we use the f-measure score, but for a product, we want the summary be as comprehensive as possible.

recall	ROUGE 1	ROUGE 2	ROUGE L
T5-small	25.997	17.113	25.512
T5-small + CNN	41.607	26.695	40.136
BERT	39.534	18.858	36.649
BERT + CNN	48.012	28.216	45.871

We see that for collection of reviews, BERT based models perform better than T5 based models.

6.5.3 Visual Inspection

While automatic evaluation metrics can be useful for evaluating summarization models, it's important to also consider other factors such as readability, coherence, and accuracy, and to supplement automatic evaluation with human evaluation and visual inspection of the summaries produced by the model.

6.5.3.1 Single Review

Those are sample summaries generated by our best model on singe reviews:

- **Original Review**: Wow! Absolutely love this!! I had a different kind but the other had one side for deposits which I could not use. This is fabulous! 2600 entry places and a place to check off for when it clears. Place to mark for tax purposes and plenty os space to wrote in. Great price! Will ALWAYS buy this now! Thanks for a fab product!
  Predicted Summary: Great price!
  Reference Summary: Best Register EVER!

- **Original Review**: Updates regularly. Fairly accurate to my experience. I got two and I was not disappointed.
  Predicted Summary: Fairly accurate to my experience
  Reference Summary: I got two and I wasn't disappointed.

- **Original Review**: Wow..this bottle was tiny. Was not sure what I expected. Still it is bleach and I cannot get too excited about that except my towels will now be so white that all the neighbors will be jealous.
- **Predicted Summary:** Fairly accurate to my experience
  Reference Summary: I got two and I wasn't disappointed.

From the inspective result, the model is able to catch most negative sentiments and details. However, most of the summaries are just repeating the beginning sentences.

6.5.3.2 Collection of Reviews

Generated summaries by best model on collection of reviews:

- **Original Reviews Set**

Predicted Summary:

- Great soundtrack I like it when he makes the guy walk like a puppet.
- love the hot nurse Joey has the hots for.
- Good, creepy and scary film. Well worth it.

Actual Summary:

- Dream Warriors rule
- Terrific- One Two, Freddy's Coming For You!

- **Original Reviews Set**

Predicted Summary:

- Good applesauce
- Delicious! I gave them one star for the non gmo baloney
- Great for small amounts

Actual Summary:

- good food, bad marketing

- Tasty and sugar free applesauce. Just right
- These are APPLE flavored. Taste great.

We see that predicted results on collections is able to catch both negative and positive sentiments from different users. In the second example, the predicted summary gives negative sentiment about NON-GMO and at the same time shows the positive sentiment.

7 Conclusion

In this project, we clean and preprocess the text data and then try T5 and BERT tokenizer to generate word vectors which will later on pass as model inputs. Then we prepare the experimental models. We build simple deep learning models such as MLP, LSTM, CNN but these models are far from enough for the summarization task. Of all the transformer body plus custom head models, T5-small + CNN is the best model for single review summarization and BERT-CNN model is the best for review collection summarization. The difference between different transformers body is small, but the difference is large after add down-stream layers. We also try to use different decoder algorithms to improve the result of the model. But there are still some problems remaining.

References

HUNG-YI LEE, *Deep Learning for Human Language Processing Online Course,* https://speech.ee.ntu.edu.tw/~hylee/dlhlp/2020-spring.php

HUNG-YI LEE, Machine Learning Online Course, https://speech.ee.ntu.edu.tw/~hylee/ml/2022-spring.php

Performance and Scalability: How To Fit a Bigger Model and Train It Faster, https://huggingface.co/docs/transformers/v4.18.0/en/performance

Stefania Cristina, A Tour of Attention-Based Architectures, https://machinelearningmastery.com/a-tour-of-attention-based-architectures/

Yang Liu, Mirella Lapata, Text Summarization with Pre-trained Encoders, https://speech.ee.ntu.edu.tw/~hylee/ml/2022-spring.php

How to generate text: using different decoding methods for language generation with Transformers, https://github.com/huggingface/blog/blob/main/notebooks/02_how_to_generate.ipynb

Annotated PyTorch Paper Implementations, https://nn.labml.ai/

Pranav Dar, 8 Excellent Pretrained Models to get you Started with Natural Language Processing (NLP), https://www.analyticsvidhya.com/blog/2019/03/pretrained-models-get-started-nlp/?utm_source=blog&utm_medium=transfer-learning-the-art-of-fine-tuning-a-pre-trained-model

Dishashree26 Gupta, Transfer Learning and the Art of Using Pre-trained Models in Deep Learning, https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/