E-Commercial Review Summarization - Abstractive Summarization

Author: Yuan Dang

George Washington University

Problem Statement

• Review summarization is the process of condensing and extracting the most relevant and informative content from some a large collection of reviews. Due to the massive amounts of user-generated content, there is a great need to reduce the text into shorter, focused summaries.

• Review summarization solves information overload, which makes it difficult for users to find relevant information amidst an abundance of words.

• Summarization provides a quick and efficient way for users to gain insights into a product or service without having to read through undreds or thousands of words.

• It can help businesses to identify areas of improvement, gather feedback, and improve customer satisfaction.

• It is not possible to manually generate these summarizations with this large data volume and hence we need automatic methods.

Abstractive vs. Extractive Summarization

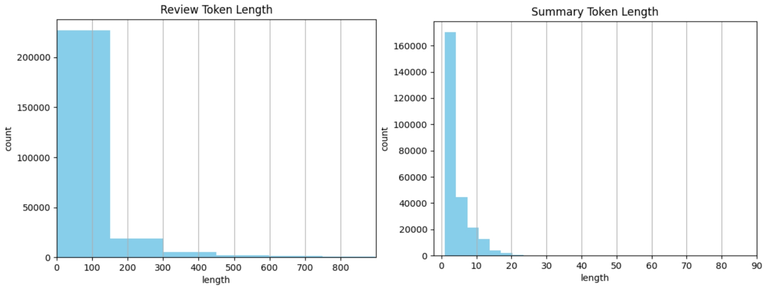
Abstractive summarization:

- shorter text keeping the most important information and sentiment at the same time
- generated by understanding the meaning of the input text

Extractive summarization:

- extract important words based on frequency
- sentiment may not be kept

Processing



Distribution of Input and Output Text Length

Approches

1. split review into words
2. find vector representation(word embedding) for each word together with its position in the sentence
3. calculate and store the similarities between word vectors with transformer for rank calculation
4. generate sentence using top-ranked words

Input

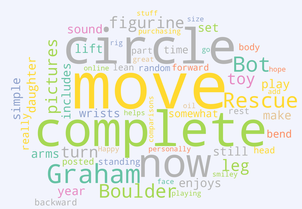
Review corpus example1

- text from one review of this book has been given as input

Input

Review corpus example2

- text from all reviews of this toy has been given as input



Output

This eBook has no pictures of the recipes.

Output

These go great playing with the Rescue Bots.

summarize long single review

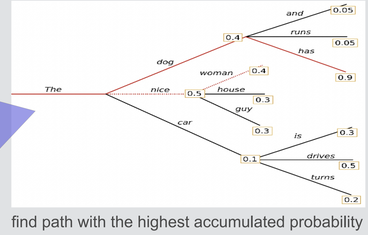
summarize a collection of reviews of single item

Sequence to Sequence Model

- Decoding Methods

beam search:

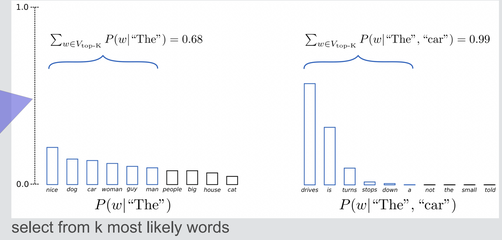
- gives the most possible word sequence
- high accuracy
  - may ignore minor information



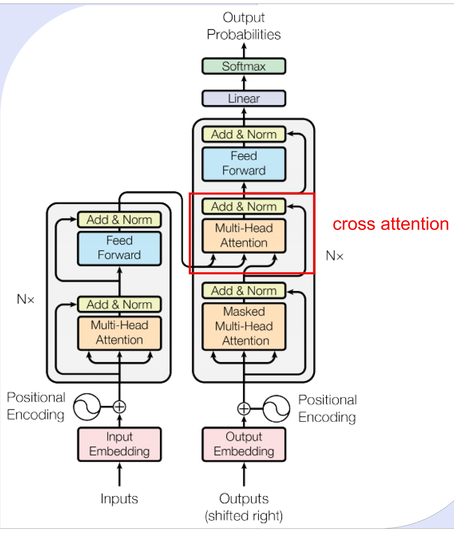
find path with the highest accumulated probability

sampling:

- adds randomness when generating next word
- catch more information
  - low accuracy



select from k most likely words



Evaluation

ROUGE Score	ROUGE 1	ROUGE 2	ROUGEL
Measure	overlap between single words	overlap between pairs of consecutive wordsv	longest common subsequence

Score	precision	recall	f-measure
Measure	How well summary is focused on capture only important information	How well summary covers all the important content	Overall balance between precision and recall

	ROUGE 1	ROUGE 2	ROUGEL
f-measure	31.071	23.091	30.077

Tips

• Use gradient accumulation to calculate the gradients iteratively in smaller batches if ran out of memory

• Disable gradient checkpoint to save memory

Reference

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

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)