Group Project Report - Group 3
Andy Christian, Rajkumar Conjeevaram Mohan, Daqian Dang
Natural Language Processing
DATS 6312
May 6, 2023

Text Summarization of News Articles for Daily/Topic News App

I. Introduction

In today's rapidly changing industries, staying abreast of the latest news, trends, and even speculations is crucial for decision makers to make well-informed choices. News articles and technical reports serve as significant sources of this information. While the intricate details contained in these documents are essential for practical implementation, it is the high-level key points that provide decision-makers with a broader understanding of the overarching landscape. A natural language processing (NLP) model adept at summarizing the main ideas of news articles and technical reports, as well as generating a comprehensive meta-summary of all pertinent information related to a specific topic, would be an invaluable asset for helping decision makers remain knowledgeable of their industry's shifting dynamics.

With this in mind, our objectives are centered around the development of a proof of concept for a news and article summary application. The prototype version of this application would be capable of ingesting a specific number of articles or reports on a given topic and generating succinct summaries for each of them. The full version would expand on this functionality by providing an overarching summary of these individual summaries, essentially offering users a one-page summary report that encapsulates the most recent trends or news in their area of interest.

To achieve these objectives, we will implement the following methodology:

1. Load and preprocess text data from a kaggle dataset, which will serve as the basis for training our summarization model.
2. Develop a state-of-the-art transformer model, such as T5-small, for generating accurate summaries of the input articles.
3. In future development phases, we will create a user-friendly application that allows users to input their desired articles or reports. The app will then leverage our trained transformer model to produce both individual summaries and a meta-summary, along with highlights of key data points.

II. Description of Data Set

The data set comes from the Kaggle page, CNN-Daily Mail News Text Summarization. It contains full articles and their summaries. According to the Kaggle page, as well as the HuggingFace page for the data set, the provided summaries were written by the human authors of the articles. There are 287,113 articles and their summaries in the training data set, 13,368 in the validation data set, and 11,590 in the test data set. The average number of tokens in an article is 781, and the average number of tokens in the human-produced summaries is 56. The data came in CSV form from Kaggle and had to be processed to send to the Transformer model.

III. Description of NLP Network and Training Algorithm

In this project, we employed the T5 model as refers Text-to-Text Transfer Transformer, introduced in the paper "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transforme" by Colin Raffel et al., (2020) is an extremely powerful transformer model that can be fine-tuned for a wide range of NLP tasks.

The T5-small model is a smaller version of the T5 model, which makes it faster and less computationally expensive while still delivering competitive performance on various NLP tasks. The architecture of T5-small is based on the Transformer architecture with some modifications. Below indicates the key characteristics of the T5-small model:

1. Model size: T5-small has approximately 60 million parameters, which makes it smaller and faster compared to the larger versions (e.g., T5-base, T5-large, etc)
2. Architecture: T5-small follows the encoder-decoder structure of the original Transformer model, with 6 layers in both the encoder and decoder. Each layer consists of a multi-head self-attention mechanism and a feed-forward network. The model also uses layer normalization, dropout, and skip connection.
3. Tokenization: T5 uses SentencePiece tokenization, which is a subword-based tokenization method that is effective for handling out-of-vocabulary words and different languages.
4. Task-specific prefix: T5 employs a unified text-to-text framework, where both input and output are considered as sequences of text. To adapt the model to a specific task, a task-specific prefix is prepended to the input sequence. (e.g., "Summarizer." for summarization or "translate English to German." for translation)
5. Pretraining objectives: T5 is pretrained using a denoising autoencoder setup, where the model learns to reconstruct the original text from a corrupted version with certain tokens masked.
6. Fine-tuning: After pretraining, T5 can be fine-tuned on specific tasks using supervised learning. This involved adjusting the model weights to minimize the

loss on the task-specific dataset. leading to high performance on a wide range of NLP tasks.

In result, the T5-small model is a smaller version of the T5 architecture, which ideally fits our project through fine-tuning.

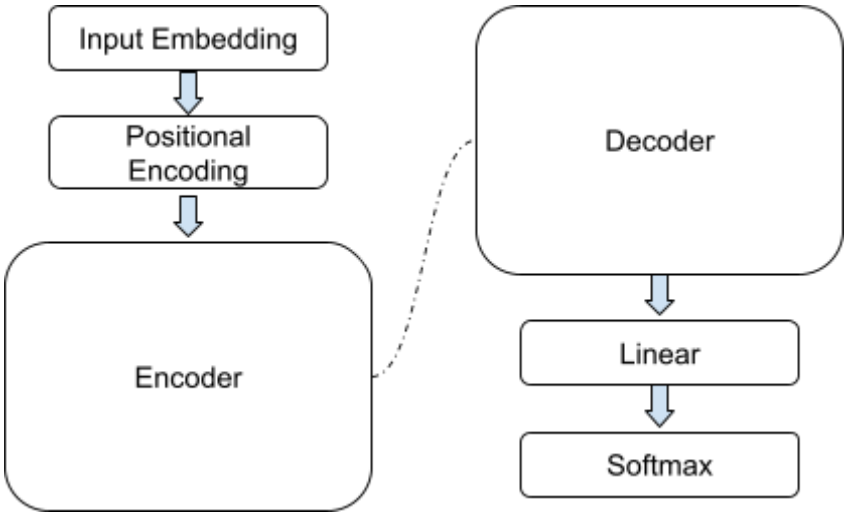
The training algorithm includes five main components (Alammar, 2018), which indicate below.

1. Input Embedding: The input text is tokenized as passed through an input embedding layer, which converts tokens into continuous vectors.
2. Positional Encoding: Input embedding is combined with positional encoding to supply the model with positional information.
3. Encoder: The T5 small encoder consists of six layers, each featuring:
    - A multi-head self-attention with 8 attention heads
    - Dropout
    - Layer normalization (incorporated after the residual skip connection)
    - A feed-forward network with a hidden size of 2048
    - Dropout
    - Layer normalization (incorporated after the residual skip connection)
4. Decoder: The T5 small decoder also consists of six layers, each containing:
    - A multi-head self-attention with 8 attention heads (decoder self-attention)
    - Dropout
    - Layer normalization (incorporated after the residual skip connection)
    - A multi-head cross-attention with 8 attention heads (encoder-decoder attention)
    - Dropout
    - Layer normalization (incorporated after the residual skip connection)
    - A feed-forward network with a hidden size of 2048
    - Dropout
    - Layer normalization (incorporated after the residual skip connection)
5. Output: The final layer of the decoder produces logits, which are then processed through a Linear layer and a Softmax activation function to generate token probabilities.

In both the encoder and decoder layers, residual skip connections are introduced after the multi-head self-attention and feed-forward networks. These connections help the model avoid the vanishing gradient problem and improve training stability. Dropout layers are positioned after attention and feed-forward network operations to regularize the model and prevent overfitting.

IV. T5 Architecture Visual Block Diagram

A visual block diagram representing the T5 architecture. Each block can represent a layer or sub-layer, and arrows can indicate the flow of data between them.



Encoder:

- Multi-head self-attention layer
- Dropout
- Layer normalization (added after the residual skip connection)
- Feed forward network
- Dropout
- Layer normalization (added after the residual skip connection)

Decoder:

- Multi-head self-attention layer (decoder self-attention)
- Dropout
- Layer normalization (added after the residual skip connection)
- Multi-head cross-attention layer (encoder self-attention)
- Dropout
- Layer normalization (added after the residual skip connection)
- Feed forward network
- Dropout
- Layer normalization (added after the residual skip connection)

V. Experimental Setup

The data from Kaggle came in three separate data sets, training, validation, and test. These three data set splits were used in our experiment. The training set was used to update the parameter weights in the different model layers after passing the inputs through the model and calculating the token-wise cross-entropy loss compared to the target. The validation set was used to evaluate how effectively the model was learning during each epoch of training by also calculating cross-entropy loss on data not explicitly used for training. The test data set was only used after the model had completed all of its training. Here, summaries produced by the model were compared to the human-written summaries by means of the rouge score. The Rouge score is an F1 score variant. The closer to 1.0 the score is, the more similar the two texts are to one another.

VI. **Rouge Metric**:

The Rouge (Recall-Oriented Understudy for Gisting Evaluation) scoring algorithm is similar to computing the F1-score that depends on Precision and Recall metrics. While the Precision here refers to the proportion of tokens in the candidate summary also found in the reference tokens, the Recall represents the proportion of tokens in the reference summary captured by the candidate summary. [5]

Rogue score enables us to understand the quality of the text summarization model by looking at the harmonic mean of Rouge's Precision and Recall metric. Rouge metric can be computed using different techniques, also called Rouge1, Rouge2, and RougeL. While the Rouge1 and Rouge2 refer to computing the Precision and Recall based on unigrams and bigrams, the RougeL is computed based on the longest sub-sequence found between the candidate and reference summary.

VII. **Optimizations**:

To improve the base model's performance, some of the hyperparameters were tweaked, and was found to perform better with the following parameters.

The Trainer API of HuggingFace allows an option to specify the number of gradients to be accumulated before updating the weights and the biases of the model. While training the model using mini-batch optimization algorithm, the parameters are updated after each step i.e., after presenting the network with a mini-batch, but using accumulated_gradient_steps replicates, at least as per my understanding, the behavior of batch optimization if the value is set to the total number of mini-batches in the training set. While this may instantaneously sound as the network is learning from a

more accurate error signal, it has its own caveats. The training set typically comes with noise, and discovering the correct value for this parameter is one of the crucial factors for faster convergence.

In terms of hyperparameters, MAX_INPUT_LEN specifies the maximum tokens of the input sentence the transformer model have visibility over, and the base model had 128. Increasing this value had positive effect on the Rouge score. Due to the resource limitations, we were unable to push beyond 512 on full training set. However, quick experiments based on the subset revealed that having a higher MAX_INPUT_LEN value offers improved performance, in terms of the accuracy - Rouge score.

Data preprocessing efficiency:

Since the data is large, it is essential to process them as quickly as possible to minimize the latency between training steps on GPU. In order to facilitate quicker processing, we used Object-Oriented paradigm, and with that in place, we used parallel calls to Dataloader, and higher prefetch value to ensure that the CPU always has a batch ready to feed the GPU in the subsequent steps.

VIII. Result

Optimized hyper-parameters:	
Learning Rate	**1e-3**
Batch size	**30**
Input Length	**1024**
Output Length	**128**
Gradient Accumulation Steps	**1**
N Parallel Calls	**10**
N Attention Heads	**8**
Query, Key, Value size	**64**
Epochs	**30**

Results at the end of 2nd epoch (full dataset):

Metrics	1024 Input Length	256 Input Length
Rouge Score	0.2013	0.054
Loss	1.05023	4.220

Results at the end of 2nd epoch (sample dataset):

Metrics	1024 Input Length	256 Input Length
Rouge Score	0.2424	0.2413
Loss	0.91646	0.633148

Sample output:

1. **Original Summary**: Experts question if packed out planes are putting passengers at risk. U.S consumer advisory group says minimum space must be stipulated. Safety tests conducted on planes with more leg room than airlines offer.

   Produced Summary: Tests conducted by the FAA use planes with a 31 inch pitch, a standard which on some airlines has decreased. Some airlines have 30 inches of space, while others offer as little as 28 inches. Some airlines offer as little as 28 inches.

2. **Original Summary**: Drunk teenage boy climbed into lion enclosure at zoo in west India. Rahul Kumar, 17, ran towards animals shouting 'Today I kill a lion!' Fortunately he fell into a moat before reaching lions and was rescued.

   Produced Summary: Rahul Kumar, 17, climbed into lions' enclosure at zoo in western India. He was next level drunk and 'thought I'd stand a good chance' against the predators. Mr Kumar had been sitting near the enclosure when he suddenly made a dash for them. He fell into a moat as he ran towards the animals and could be rescued.

3. **Original Summary**: Nottingham Forest are close to extending Dougie Freedman's contract. The Forest boss took over from former manager Stuart Pearce in February. Freedman has since lead the club to ninth in the Championship.

Produced Summary: Dougie Freedman set to sign new deal at Nottingham Forest. Freedman has stabilised Forest since replacing Stuart Pearce. Freedman has impressed at the City Ground since replacing Stuart Pearce.

4. **Original Summary**: Fiorentina goalkeeper Neto has been linked with Liverpool and Arsenal. Neto joined Firoentina from Brazilian outfit Atletico Paranaense in 2011. He is also wanted by PSG and Spanish clubs, according to his agent. CLICK HERE for the latest Liverpool news.

   Produced Summary: Fiorentina goalkeeper Stefano Castagna also wanted by PSG and PSG. Brazilian goalkeeper expires in June. Simon Mignolet was dropped from the club earlier in the season. Neto has not featured for the senior side since joining in 2011.

5. **Original Summary**: Tell-all interview with the reality TV star, 69, will air on Friday April 24. It comes amid continuing speculation about his transition to a woman and following his involvement in a deadly car crash in February. The interview will also be one of Diane Sawyer's first appearances on television following the sudden death of her husband last year.

   Produced Summary: Jenner, 65, will speak in a 'far-ranging' interview with Diane Sawyer on April 24. The interview comes amid growing speculation about the father-of-six's transition to a woman. Bruce recently made headlines for his involvement in a deadly car crash in California in February.

IX. Summary and Conclusions

The project was successful in creating a proof of concept for using a transform to do summarization of news articles for a Trending News/Academic Report Summarizer App. Using a pretrained model, we were able to fine-tune it to the data set and produce summaries of new articles. However, working with transformers is very computationally expensive. Even using our AWS instance and small batch sizes, training the full set of 200k+ inputs would take many hours. This makes experimenting and optimizing very slow to do. With these challenges, the summaries produced were of varying quality. The challenge goes beyond considering only the loss and rouge scores of a produced summary. For example, a produced summary that matched the target for 50% of the words, would get a decent score, but if the non-matching 50% of the produced summary was not “top-level” information but some detail, then it would not really be a good summary. The first produced summary in the examples above is a good example of this.

Additional optimization and training time could certainly help resolve this, however, taking such time may not be entirely feasible for a small at-home development context. Thus, this experience has motivated us to consider some additional strategies to achieve the ultimate goal of creating an app that can create a 1 page summary of N number or news articles on a given topic. For example, in order to help “guide” the model to pick out “top-level” summary data, perhaps each paragraph of an article could be used as input, producing a one sentence summary of the paragraph. Then those paragraph summaries could be concatenated into the summary of the full article. If this was still too detailed or lengthy, then that collection of paragraph summaries could be summarized. This would help ensure that the full article was covered and that more critical details were included in the summary.

It would also be interesting to create a second classification model, which used zero-shot classification to see whether the full article, human-written summary, and model-produced summary for each input were all classified the same. While this might seem trivial if used for articles all of different domains, if the articles were all about a single main topic, the categories given to the zero-shot classification model could help distinguish certain views, positions, conclusions, details, etc. Using it in this manner, it could become a helpful tool to see how well the summarization model was working. Taking the first example summary from the results shared above, one could test if the produced summary was classified as being about passenger risk. The produced summary would fail this test as it focused on minute details and not the main point of the article (passenger risk due to small seat space/legroom).

References:

1. *CNN-DailyMail News Text Summarization*. (n.d.). CNN-DailyMail News Text Summarization | Kaggle. https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail
2. *cnn_dailymail · Datasets at Hugging Face*. (2023, January 24). Cnn_Dailymail · Datasets at Hugging Face. https://huggingface.co/datasets/cnn_dailymail
3. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv preprint arXiv:1910.10683. https://doi.org/10.48550/arXiv.1910.10683
4. Alammar, J. (2018, June 27). *The Illustrated Transformer*. The Illustrated Transformer – Jay Alammar – Visualizing Machine Learning One Concept at a Time. http://jalammar.github.io/illustrated-transformer/
5. Alvin, T. P. (2022, March 21). *Introduction to text summarization with Rouge scores*. Medium. Retrieved May 5, 2023, from https://towardsdatascience.com/introduction-to-text-summarization-with-rouge-scores-84140c64b471