CS F429 - Natural Language Processing

A Project Report on

Hybrid Summarization of Emails using Synthetic Dataset

Contents

Contents 1
Abstract 1
Task Definition 2
Dataset 2
Model 3
Parameters 3
Loss Function 4
Results 4

Abstract

Text summarization is a common challenge in Natural Language Processing, with three primary approaches: Extractive, Abstractive, and Hybrid Summarization. In Extractive Summarization, sentences are ranked based on their relevance to the document's main meaning, often using techniques like TF-IDF or Graph-based methods. Models such as T5 and Seq2Seq employ these methods. Conversely, Abstractive Summarization generates new sentences that capture the essence and details of the original text, with models like Bard and Flan-T5 being popular choices. Hybrid Summarization combines elements of both approaches, with tools like PreSumm and Pointer-Generator. Our goal is to fine tune a summarization model tailored for emails, and observe its performance against a synthetically generated dataset. We present this project as a proof of concept. The unique aspect of our work lies in the dataset we would generate for training. We utilize the pre-trained models mentioned earlier to generate summaries for an email dataset. These models, including Google AI LLM, LangChain and Flan-T5 cover a mix of Abstractive, Extractive, and Hybrid methods. We then employ a Sentence Ranker to combine the outputs generated by these models. Why do we need a synthetic dataset? The reason is the absence of suitable datasets for this specific task. Since each pre-trained model aims to produce realistic outputs, we can consider using an ensemble method that combines their results to get closer to the actual value. Our model is trained on this synthetic dataset, and we compare its results with the results generated by the pre-trained models. The email dataset we use for this project is aeslc, readily available on Hugging Face. This model can be a valuable tool for users dealing with lengthy emails, potentially identifying spam and providing a concise overview of the email's content, saving time and effort.

Task Definition

Fine tuning summarization models for emails using supervised learning techniques on synthetically generated dataset.

Dataset

- Raw Dataset
  The initial raw dataset that was used for this training is the aeslc dataset available on HuggingFace. The dataset consists of a variety of email exchanges between people or parties associated with the Enron Corporation, along with their respective subject lines. The subject lines were not used as part of the training because they were too small and in most of the cases provided little to no information about the actual email itself.

- Pre-processing

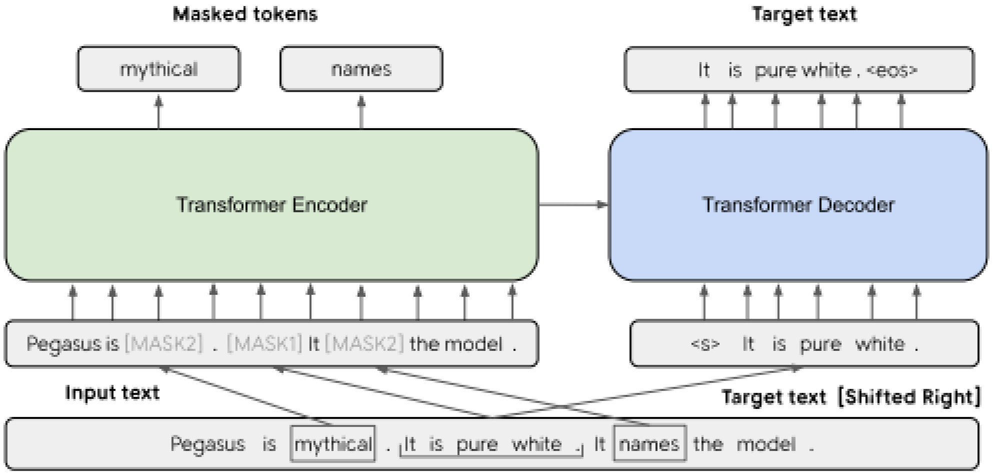
  Text cleaning:
  1. Subjects and annotations from the email have been removed, since the subjects are small and provide no real meaning about what the email is trying to convey.
  2. Greetings and pleasantries have been removed from the email because they provide no useful information and only increase computation.
  3. Decorative text and characters that cannot be understood by models (like Unicode) have also been removed.

  NLP based Preprocessing:
  1. Paragraphs within an email are tokenized into sentences using the nltk library in order to generate sentence embeddings which do not depend on document level encoding.
  2. Sentences with similar semantic meaning are not required in the corpus because they will lead to extra computation. We get rid of semantic duplicates by running a cosine similarity check on the embeddings of two sentences within an email, and pick one if similarity > 0.75 (adjustable hyperparameter).
  3. Stop words haven't been removed because we would be working on a combination of abstractive and extractive summarization. For extractive summarization, we require full sentences to get well-defined summaries.

- Target Summaries Dataset
  A hybrid summarization (abstractive followed by extractive summarization) method was used to generate summaries for the preprocessed emails. Bard and Google FLAN-T5 are the models we chose to perform abstractive summarization, and TextRank was chosen to perform extractive summarization.
  The preprocessed emails were passed as input to Bard and Google FLAN-T5 to first generate an abstractive summary. Then, the TextRank model was used to extract the most important sentences from this abstractive summary. The resultant dataset was used as a benchmark to evaluate the summaries generated by the fine-tuned PEGASUS model.

Our final dataset set is split into 3 parts: Train set, Dev set and Test set. The train consists of 6847 training examples

Model


Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

The base architecture of PEGASUS is a standard Transformer encoder-decoder. What makes this model different is not the architecture, but the pre-training objectives. The paper uses two techniques: Gap Sentences Generation and Masked Language Model. During GSG, the top m most important sentences are taken from the document (to be appended to final summary later), and their corresponding positions are masked in order to create a predictive model simulating abstractive summarization. For MLM, they select 15% tokens in the input text, and the selected tokens are 80% of time replaced by a mask token, or 10% of time replaced by a random token, or 10% of time unchanged. These masks are then predicted during training and the result is evaluated against the final target summary. The encoder is responsible for processing the input text, extracting its essential meaning, and encoding it into a context vector. This context vector serves as a concise representation of the input, capturing the key information and relationships between the sentences. The decoder, tasked with generating the summary, takes the context vector as input and progressively decodes it into a sequence of words. At each step, the decoder attends to the context vector and the previously generated words to produce the next word in the summary.

Parameters

Trainable Parameters:

Since the Pegasus model follows a normal transformer (encoder - decoder) architecture, the parameters trained are the same as for any basic transformer architecture. The parameters are as follows:

- Word embeddings: To represent the meaning / context of the words in the sentences.

- Positional encoding: We attach positional embeddings to word embeddings in order to preserve positional context.
- Encoder layer weights: To capture the context of the sentences, we use an encoder layer.
- Decoder layer weights: The decoder layer generates the summary on the basis of the context passed onto it by the encoder.
- Attention mask parameters: In order to get the most relevant parts of the sentences while capturing context, attention masks are used within the encoder as well as the decoder.
- Fully connected layers: The outputs of the encoder - decoder are passed to fully connected layers, which will have their own parameters. This layer is added to refine the outputs.
- Additional bias: At every output layer, there will be some bias terms which will have their own weight.

Hyper-parameters:
At each step, there are certain parameters that can be changed in a training run. These are not limited to the parameters that are used during training. We start off with parameters used during the generation of the dataset. Here, we set the number of sentences in the generated summary to be a maximum of 3. For the LLMs, we tried different prompts to generate the best summaries, and finally we went with the prompt mentioned in the code (following a prompt template suited to each LLM). Coming to the model architecture, the tokenizer length is set to a max of 1024. The number of epochs is set to 1, the weight decay (α) is set 0.01, the model is evaluated at every 500 steps and the gradient is accumulated at every 16 steps.

Loss Function

The PEGASUS model uses a cross-entropy function to calculate the loss and backpropagate through the neural network to update its weights. The PEGASUS model generates a probability distribution over the vocabulary for each position in the output sequence (summary). The cross-entropy loss is then computed by comparing this distribution to the true distribution, which is typically a one-hot encoded vector representing the actual words in the target summary.

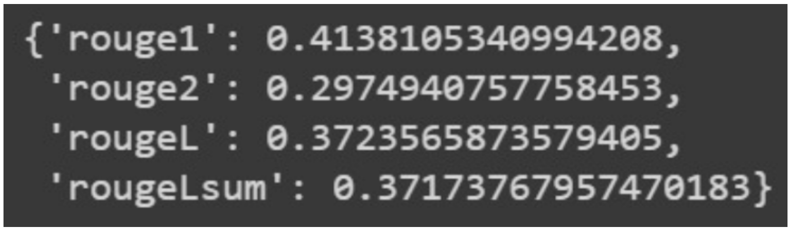
$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)}$$

Results

We used the ROUGE score metric to evaluate how close the generated summary is to the actual email. ROUGE-N works on the basis of the number of overlaps of n-grams between the system and reference summaries, whereas ROUGE-L works on the longest common subsequence of words between the generated summary and the actual text.

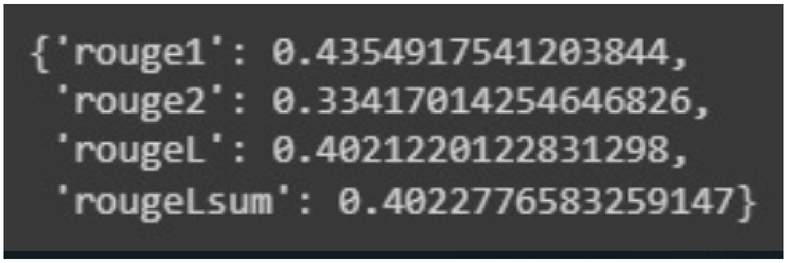
The ROUGE scores were calculated twice, once without fine-tuning the PEGASUS model, and then later by fine-tuning it with our training dataset to observe the fine-tuned model's performance.

ROUGE scores before fine-tuning:



```
{'rouge1': 0.4138105340994208,
 'rouge2': 0.2974940757758453,
 'rougeL': 0.3723565873579405,
 'rougeLsum': 0.3717376795747018З}
```

ROUGE scores after fine-tuning:



```
{'rouge1': 0.4354917541203844,
 'rouge2': 0.3341701425464682б,
 'rougeL': 0.4021220122831298,
 'rougeLsum': 0.4022776583259147}
```

Summarization results for emails:

```
Email
Please find attached the latest , and what should be the final for the immediate period of time ,
copy of the marketing list . Please filter the PA column by your name to double-check against the
list you are currently working off of . There are some smaller subsids of larger companies
previously assigned now listed . I pulled these subs in from the credit pre-approval list . I
will be passing out draft contracts folders for these counterparties tomorrow to the people
responsible for the respective parent companies . There are some companies on your lists that do
not have draft packets . These companies have NOT been pre-approved by credit , but credit is
working on them now . Continue to call these customers regarding password applications , but do
not promise that the ISDA documents , etc . are immediately forthcoming until we receive those
worksheets from credit . Please continue to pass on both the pa information and the draft packets
as you receive the correct contact info . If you have any questions at all , please call me at
5-3614 or come by my desk at 06541 in the rotunda . Thank you ,

Expected Summary:
Latest Marketing List - Please double-check against the list you are currently working off
of.Some smaller subsids of larger companies previously assigned now listed.Continue to call these
customers regarding password applications, but do not promise that the ISDA documents, etc.

Model Summary:
Please find attached the latest, and what should be the final for the immediate period of time,
copy of the marketing list.Please continue to pass on both the pa information and the draft
packets as you receive the correct contact info.
```

Email
Susan : I 'm not sure that what I told you to do in the CSA Annex Custodians and their qualifications work so for now let 's keep it out and go back to our old language . Sara and Mark : Shari had a counterparty point out to her that in our CSA Annex language where we talk about the circumstances under which a party can `` hold '' collateral , all of our triggers are based on something happening to our counterparty and nothing is said about what would happen if a Custodian holds the colateral and fails to be creditworthy . Do you have any thoughts on this ? At first I thought that we should require the Custodian to be a Qualified institution but I 'm not sure that that is the right approach . Let me know what you think . Carol

Expected Summary:
Susan : I'm not sure that what I told you to do in the CSA Annex Custodians and their qualifications work

Model Summary:
Shari had a counterparty point out to her that in our CSA Annex language where we talk about the circumstances under which a party can hold '' collateral, all of our triggers are based on something happening to our counterparty and nothing is said about what would happen if a Custodian holds the collateral and fails to be creditworthy.

Email
, speaking with me the other day regarding KWI . I am writing to update you on the progress of this search . Our client is in the final stages of negotiations with an excellent candidate . lot for

Expected Summary:
KWI is in the final stages of negotiations with an excellent candidate......

Model Summary:
The client is in the final stages of negotiations with an excellent candidate.

Email
It 's official ! Dawn and Gary Wilson are the proud parents of a beautiful baby girl . She was born at 10:41am on 9-20-01 , and weighed in at 6lbs , 13oz . They named her Grace Elizabeth . Mom and baby are doing great and the proud father delivered the news at 1:00pm . Dawn is registered at Texas Womens Hospital under the last name of Kenne . I 'm sure you will all join me in wishing the new family lots of joy !

Expected Summary:
Dawn and Gary Wilson are expecting a baby girl.They named her Grace Elizabeth.They are doing great and the proud father delivered the news at 1:00pm.

Model Summary:
Dawn and Gary Wilson are the proud parents of a beautiful baby girl.She was born on 9-20-01, and weighed in at 6lbs, 13oz.They named her Grace Elizabeth.

```
Email
Mark - I just wanted to update you regarding this Thursday 's program with the ADL and Judge
Bobby DeLaughter . Those in the legal profession can receive 1 CLE for attendance . sending out
the memo last week to Enron 's legal Laura

Expected Summary:
Thursday's program with the ADL and Judge Bobby DeLaughter.Those in the legal profession can
receive 1 CLE for attendance.

Model Summary:
Thursday's program with the ADL and Judge Bobby DeLaughter.Those in the legal profession can
receive 1 CLE for attendance. sending out the memo last week to Enron's legal Laura.
```

Group Members:

1) Arkishman Ghosh (2020A7PS2077H)
2) Sriram Balasubramanian (2020A7PS0002H)

Github Repository