FeedBack Summarization

Sections:

1. Data Analysis
2. Model Training
3. Evaluation
4. Future Improvements
5. API

DATA ANALYSIS

1. Total 5009 samples in the data.
2. 1008 samples had “**NA**” as Summary. Many samples had variations of **None** in Summary. Eg: “\n none”, “\n\n\nNone” etc. To standardize such summaries, I stripped the Summary and replaced all such occurrences of **None** with “**Nothing**”**.** This was done because **“None”** gets saved as “NA” in pandas. After preprocessing, number of samples with “**Nothing**” summary is 1069. That is around **20%** , quite significant.
3. **Notion** had the most number of samples while **Zoom** had the least [Fig 1]. Most of the text belonged to **Twitter** and least to **G2** [Fig 2]

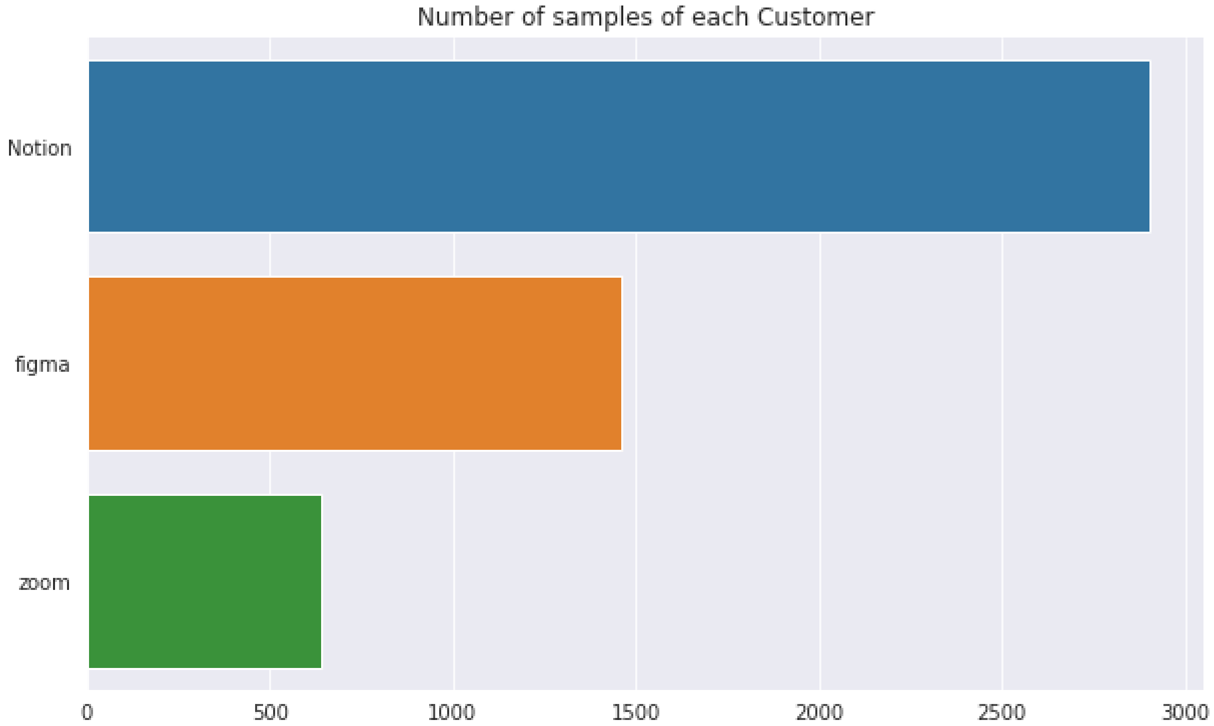

Number of samples of each Customer
Notion
figma
zoom
0
500
1000
1500
2000
2500
3000

Fig 1

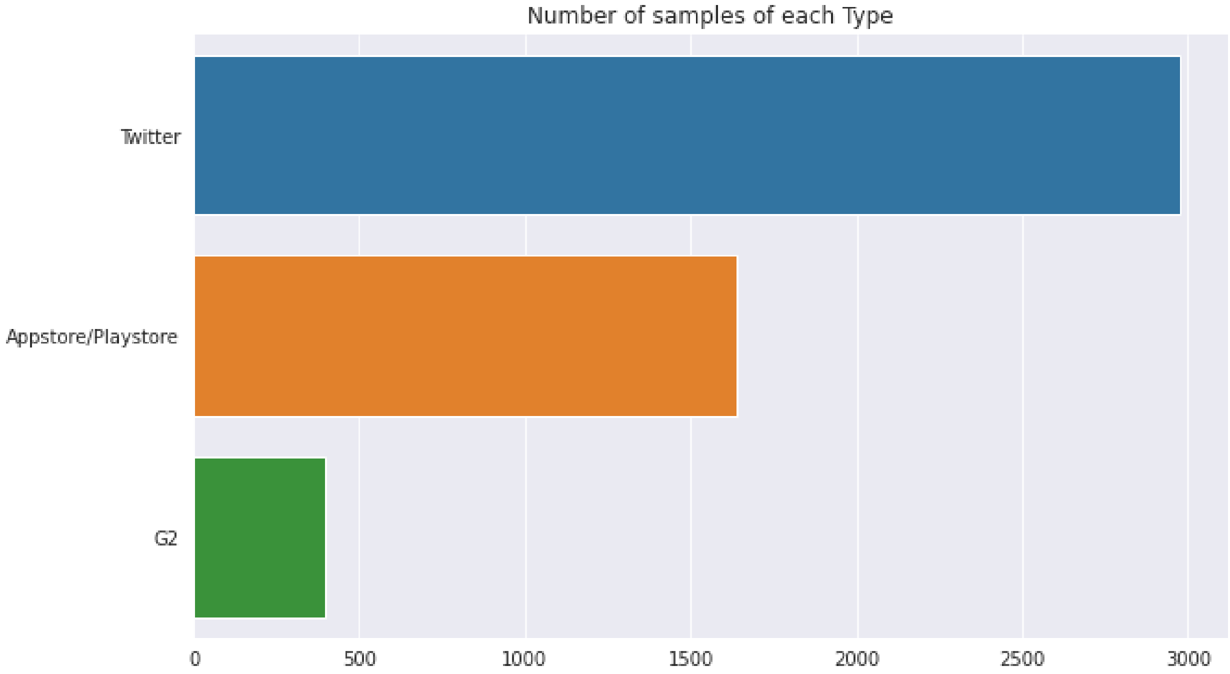

Number of samples of each Type
Twitter
Appstore/Playstore
G2
0
500
1000
1500
2000
2500
3000

Fig 2

4. Intuitively, summary tends to be shorter than text in most cases. So I plotted the ratio **of length of Summary to length of Text** [Fig 3]. We see that many summaries are much larger than the text. So we need to look at some samples .

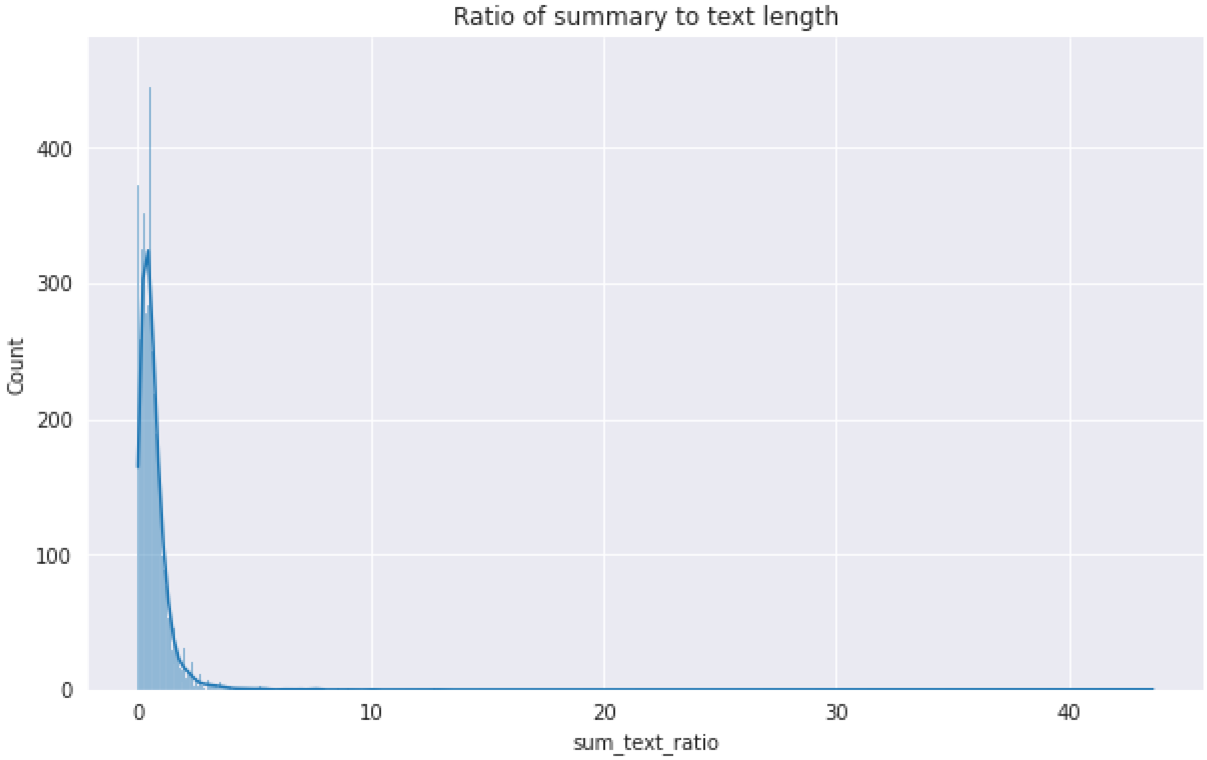
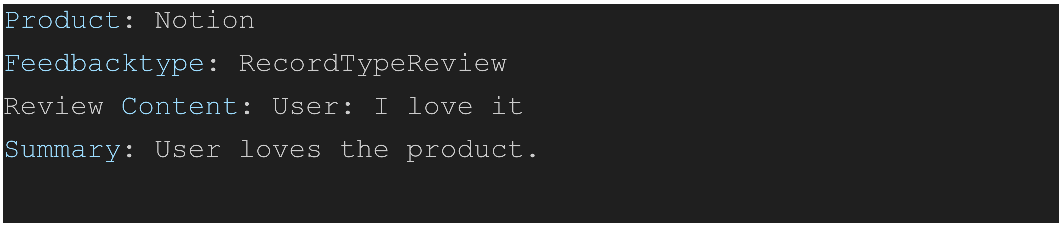


Fig 3

5. After looking at some high ratio samples, I realized the following points:
    a. Twitter has more samples with large summary-to-text ratio [Fig 4]
    b. Format of some summary was different than the usual format. It had keys like Product, FeedbackType, Review Content, Summary. Need to only extract information in Summary section. Eg:



```
Product: Notion
Feedbacktype: RecordTypeReview
Review Content: User: I love it
Summary: User loves the product.
```

    c. Many Twitter samples had <STRICT_LINK> tag which maybe had more information about the tweet and so Text was smaller than Summary. Eg:

d.
```
@NotionHQ kidding? <STRICT_LINK>
```

e. Sometimes, the Text was in a different foreign language where the words maybe written without space in between (I was splitting using space separator). And the Summary was written in English. So Text was smaller than Summary. Eg:
```
User: iPhoneとiPadだと日本語でタイトル入力すると消える！
```

f. Couple of words expanded to sentence in Summary. Eg:
```
Text: User: Network issue
Summary: User experienced network issues while using Zoom.
```

6. Some Help or Complaint kind of reviews had no Summary. Eg:
```
'# Following is `Help` or `Complained` category
#'User: Have you ever experienced this situation? The payment
interface still pops up when you subscribe and then use it'

# `Help` category
# 'User: I HAD SUCH A LONG NOTION PAGES AND MY ACC GOT REMOVED FOR
WHAT???? HELP'
```

7. Checked for duplicate Text. Found **53 Text** samples having more than 1 occurrence. Following text had the highest occurrence of **37**. **Falcon** paper showed that data **deduplication** is a very important step. So, I dropped Text duplicates and only kept the first occurrence.
```
User: RT @AdhamDannaway: ⭐ Learn Design\nA free course by @Figma to
help you get started in design.\n◆ 12 lessons\n◆ 5 Exercises\n◆
Practice files..
```

8. Obviously, in Summary **Nothing** has the most repetition, followed by following sentences:
```
# User is excited about the new Notion app and is looking forward to
using it.
# User is excited about the new features in Notion and is looking
forward to using them.
# User is excited about the new features in the latest version of
BrowserCO, including the ability to add a custom logo and customize
the color of the logo.
```

9. Closely analyzed some Twitter samples, because they had a lot of <STRICT_LINKS> earlier. Realized that the Summary of many samples is unrelated to the corresponding Text. Found many samples in G2 as well. Maybe the latter part of the dataset was intentionally or unintentionally shuffled? Eg:

```
# Text:
# User: @nick__pattison @figma That’s worrying.
# Summary:
# A user shared their DonorsChoose project for books about taking
care of the Earth and suggested that students could use Figma's
FigJam to collaborate on ideas and action plans based on their
readings.

# Text:
# User: Why does @NotionHQ use a Somali domain?
# Summary:
# User is frustrated with the lack of updates on the Notion website
and mentions that they have been waiting for a new feature for a
long time.

# Text:
# User: Here’s an invite to Arc, the browser. Use it, you'll love
it!
#arc @arcinternet <STRICT_LINK>
# Summary:
# User is excited about the new features in the latest version of
Firefox, including the ability to create a new tab with a custom
background and a new feature that allows users to create a new tab
with a custom background and a new tab page.
```

10. Since many Summaries looked unrelated to the Text, I thought of calculating similarity scores of each Text-Summary pair. Used **SentenceTransformer** model **all-MiniLM-L6-v2** to create **Embeddings** of Text and Summary and then calculated **Cosine-Similarity.** Following is the distribution of similarity scores [Fig 4]. **Majority** of the similarity scores are less than **0.5** . Certainly, the Text and Summary are **mismatched** in significant samples.

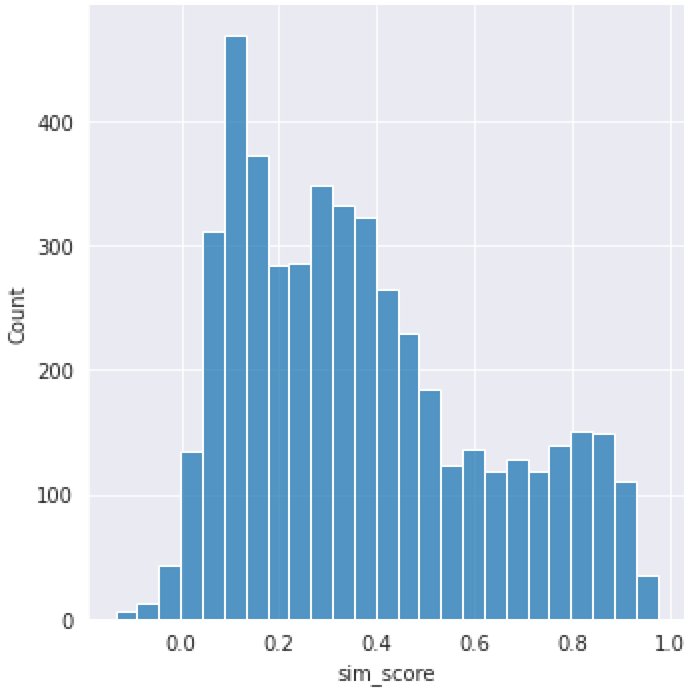


Fig 4

11. To mitigate the above issue, I thought to use **Approximate Nearest Neighbor Search** to match the Text to the Summary. Since this is a **many to one** problem, i.e, many Text samples can have the same Summary, I indexed the Summary embeddings using **FAISS**. Then for each Text sample I found the approximate nearest Summary neighbor.

    NOTE: If we index the Text embeddings and try to find the nearest Text neighbor for each Summary, then for each repetitive Summary, we get the same Text as the nearest neighbor and thus we lose context of the **many to one** problem.

12. I was concerned about **Nothing** summary samples. I presumed that they would be matched with very few or no Text samples. So, I compared the **original** Summary and the **ANN** Summary for the samples which had the original Summary as Nothing. Found that for extremely short (< 4 words) Text samples, the original Summary (i.e **Nothing**) suited more than the original Summary. So replaced the ANN Summary with **Nothing** for such Text samples.

13. Plotted the similarity scores between this new Summary and Text pairs [Fig 5]. Certainly, the Text-Summary pairs look more related than the original data.

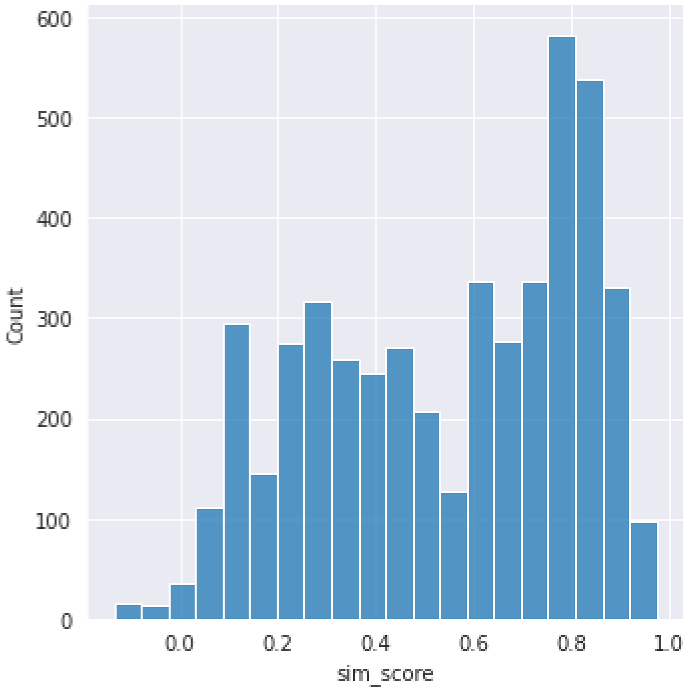


Fig 5

14. Three more observations were done while reading the Text and Summary.
    a. The Twitter samples had a lot of emojis, special symbols and STRICT_LINKS
    b. Almost all Text samples started with **User:**
    c. Some Summary samples had **different format** as written in point **5b**
15. So I decided to do two experiments to compare the performance of model on:
    a. Preprocessed data
    b. Non preprocessed data
16. Shuffled and created a 70-10-20 split for the train-val-test.

MODEL TRAINING

1. To keep it simple, I am training a Flan-T5 model for summarization. Here I thought to do four experiments:
    a. Full model fine-tuning of **Flan-T5-small** on processed data
    b. Full model fine-tuning of **Flan-T5-small** on unprocessed data (Point 15 from Data Analysis section)
    c. LoRA with **Flan-T5-large** on processed data
    d. LoRA with **Flan-T5-large** on unprocessed data
2. Used **Huggingface** for training and evaluation
3. **Flan-T5** needs **"summarize:"** at the start of each

4. Obtained max length for Summary and Text using all splits. Padded the Input tokens for max length and made their corresponding label as **-100** because we are neglecting **-100** for loss computation.
5. Evaluated the model regularly on the validation set while training. Logged the loss and metric scores for train and val sets on **Weights and Biases**.
   [**Click here to see the graphs**]
6. I see that the train and eval loss are decreasing, thus the models are getting trained and there are no signs of overfitting [Fig 10]. These are loss curves for the 4 experiments. Solid lines represent train loss and Dashed lines represent eval loss

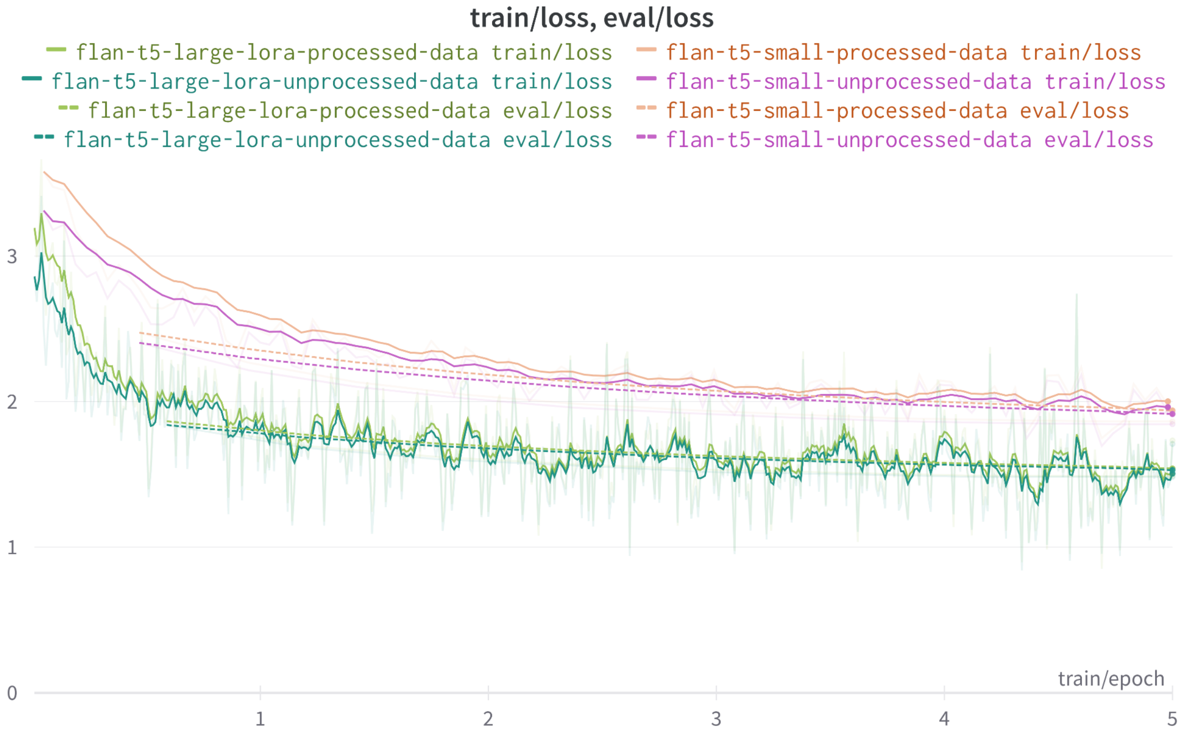


Fig 10

7. With every evaluation on the validation set, rougeLSum is increasing indicating that the model is performing better on unseen samples. [Fig 11]

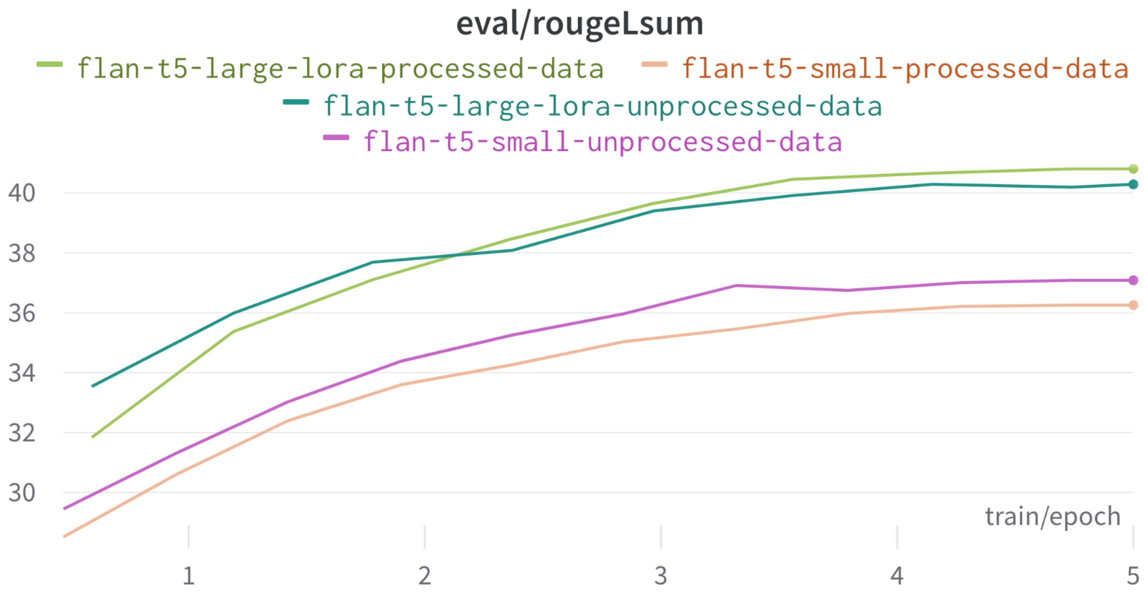


Fig 11

8. Models can be trained and used for prediction using following commands from the **code directory**

```
// flan t5 small on unprocessed data
python3 train.py --dataset_path ../data/final_ann_data --run_name
flan_t5_small_non_preprocess --log --batch_size 16

// flan t5 small on processed data
python3 train.py --dataset_path ../data/processed_final_ann_data --run_name
flan_t5_small_preprocess --log --batch_size 16

// flan t5 large on unprocessed data with LoRA
python3 train.py --dataset_path ../data/final_ann_data --run_name
flan_t5_large_peft_non_preprocess --use_peft --model google/flan-t5-large --log
--batch_size 2 --eval_log_steps 1000

// flan t5 large on processed data with LoRA
python3 train.py --dataset_path ../data/processed_final_ann_data --run_name
flan_t5_large_peft_preprocess --use_peft --model google/flan-t5-large --log --batch_size
2 --eval_log_steps 1000
```

EVALUATION

1. Test set metrics. **Flan-T5-Large with LoRA on Preprocessed data performs the best.**

Model Name	**Dataset**	**Rouge1**	**Rouge2**	**RougeL**	**RougeLSum**
Flan-T5-Small	Unprocessed	0.4157	0.2680	0.3797	0.3841
Flan-T5-Small	Preprocessed	0.4022	0.2564	0.3679	0.3717
Flan-T5-Large LoRA	Unprocessed	0.4458	**0.3089**	0.4123	0.4166
Flan-T5-Large LoRA	Preprocessed	**0.4494**	0.2960	**0.4170**	**0.4206**

2. Click here for predictions. The predictions on test set using LoRA Flan-T5-Large
3. Used **Rouge1, Rouge2, RougeL, and RougeLSum** for evaluation. This falls under the category of **n-gram metrics [**Fig 6**]**

rouge1	**0.4494**
rouge2	**0.2960**
rougeL	**0.4170**
rougeLsum	**0.4206**

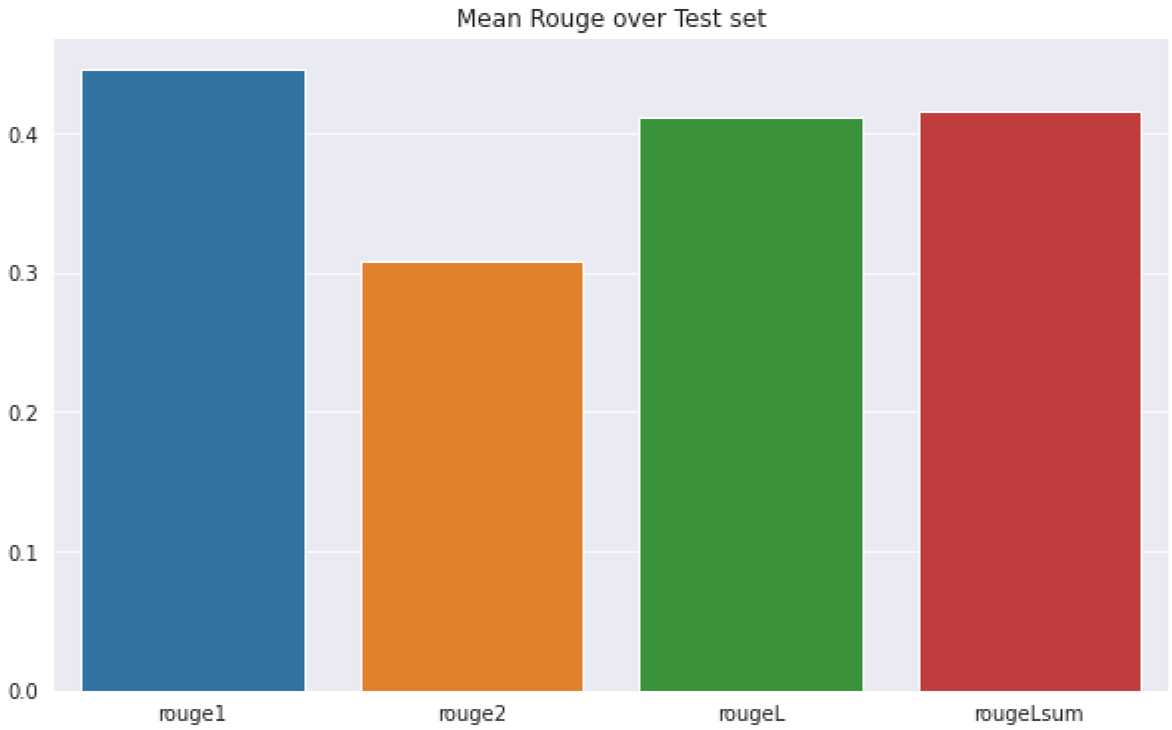


Fig 6

4. Can use **similarity score based metrics**. Find cosine similarity between GT and prediction using **BertScore**. This can take care of changes in structure of predicted language but same semantics.
5. Metrics are calculated separately **Customer**-wise [Fig 8], **Type-**wise [Fig 7] to know on which Customer and Type model fails the most. We see that the model underperforms on **G2 Type** and **Figma Customer.** We will look into the samples below.

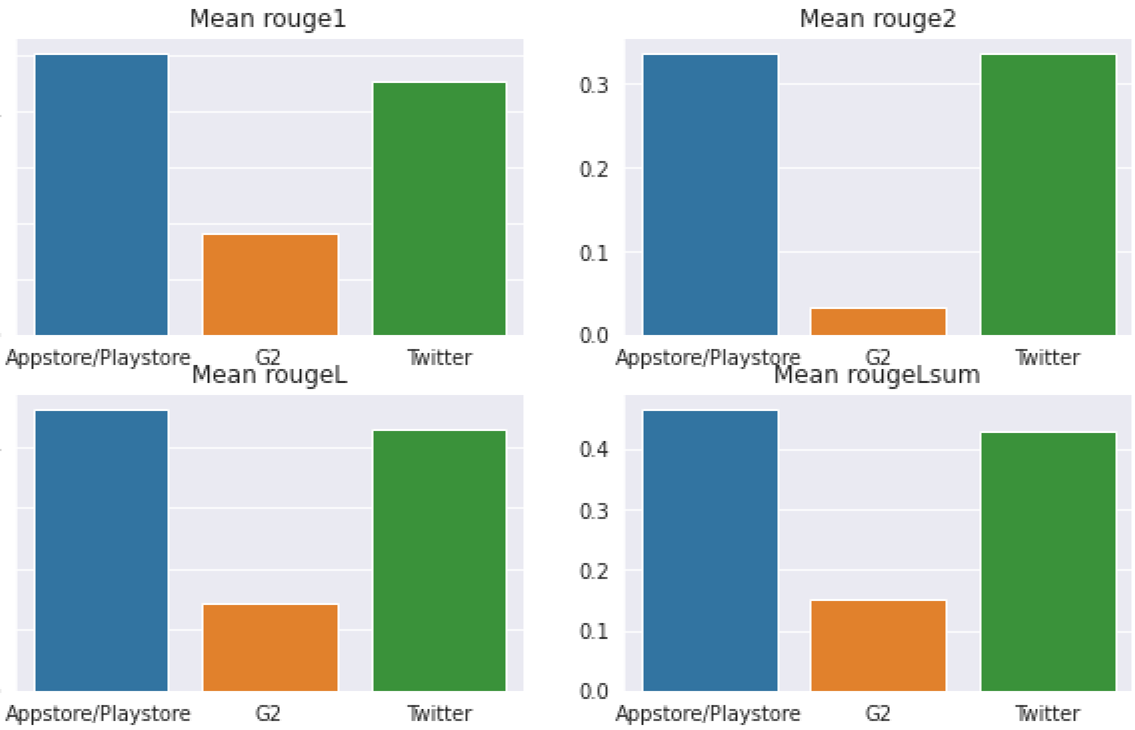

Mean rouge1
0.0
0.1
0.2
0.3
0.4
0.5
Appstore/Playstore
G2
Twitter
Mean rouge2
0.0
0.1
0.2
0.3
Appstore/Playstore
G2
Twitter
Mean rougeL
0.0
0.1
0.2
0.3
0.4
Appstore/Playstore
G2
Twitter
Mean rougeLsum
0.0
0.1
0.2
0.3
0.4
Appstore/Playstore
G2
Twitter

Fig 7

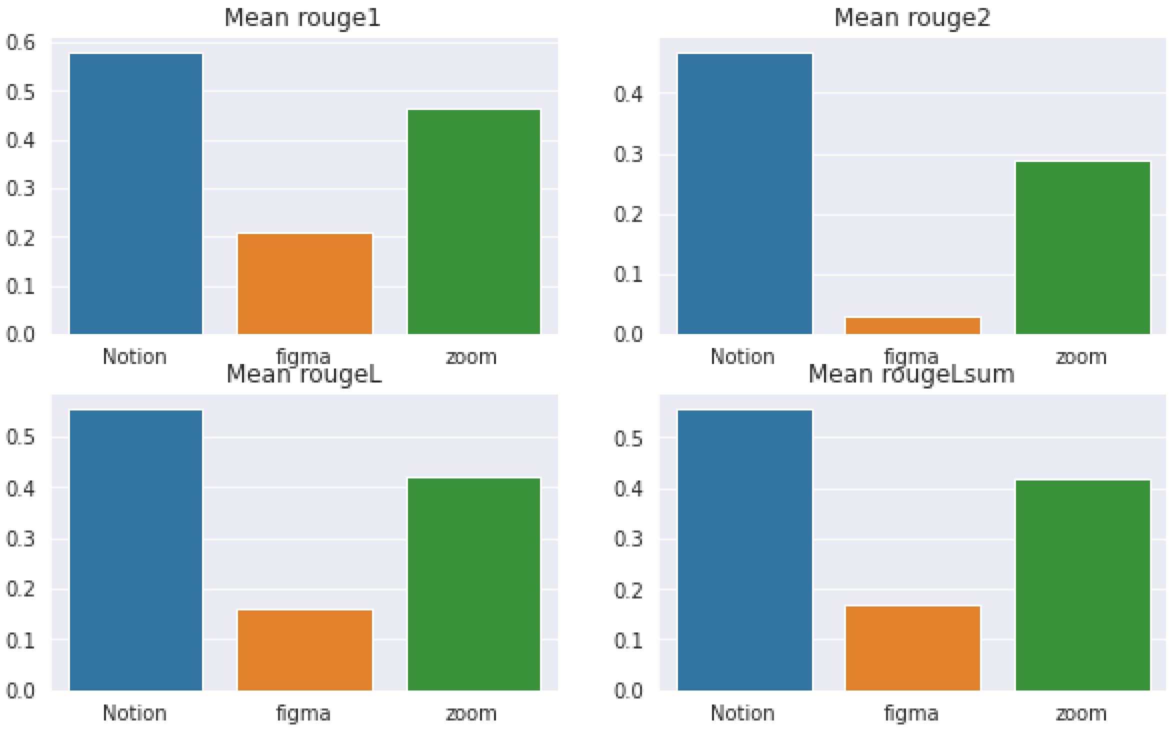

Mean rouge1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Notion
figma
zoom
Mean rouge2
0.0
0.1
0.2
0.3
0.4
Notion
figma
zoom
Mean rougeL
0.0
0.1
0.2
0.3
0.4
0.5
Notion
figma
zoom
Mean rougeLsum
0.0
0.1
0.2
0.3
0.4
0.5
Notion
figma
zoom

Fig 8

6. Scatterplot of similarity score between **Text-GT_Summary vs Rouge.** There is a **positive correlation** between this similarity score and Rouge. This suggests that if the Summaries are made more similar or appropriate to the Text, then the model can learn better. Infact, it is possible that the **model has predicted more related** Summary to Text than the GT Summary and so the rouge score is low for low similarity score samples.

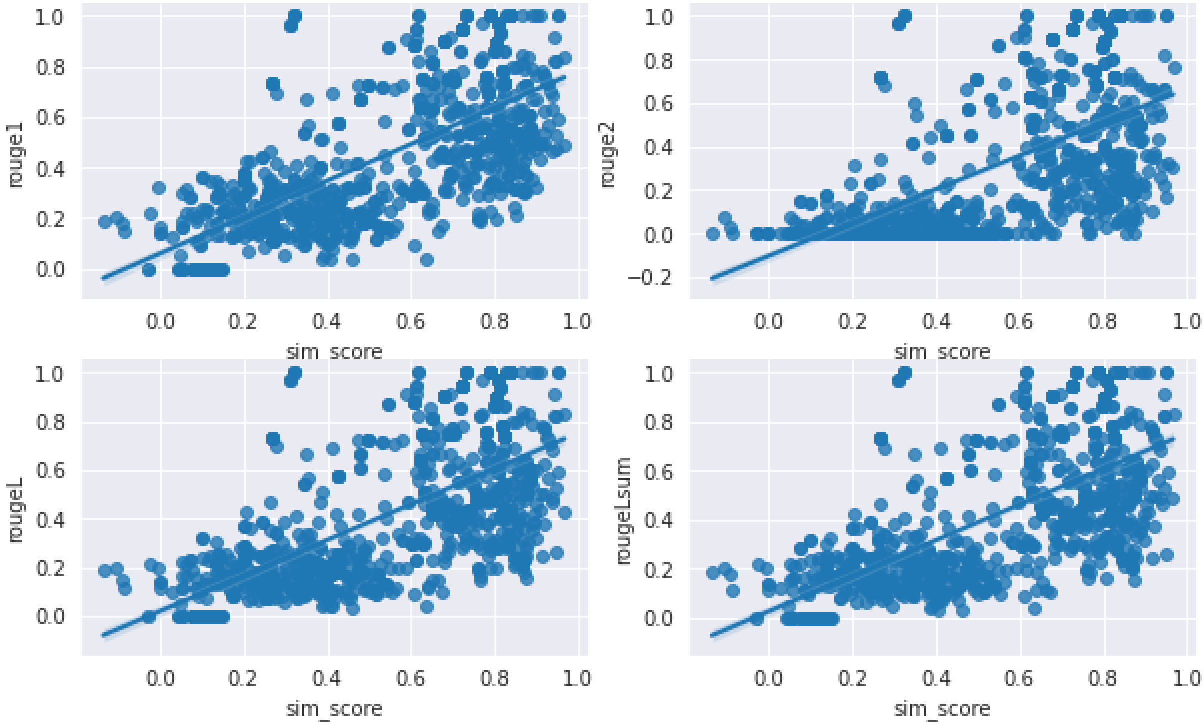


Fig 9

7. Manually looked at Text, Summary and Prediction for samples having low metric scores. Lets look at G2 + Figma first. Below are 6 samples with **Text, GT** and **Prediction**. In all the 6 samples, the **prediction is better than the GT.** Only problem is that the model **sometimes generates incomplete sentences and stops abruptly**. Eg: **4 below.**

1. **Text**:

User: Before starting each project, I meet with my clients to know what they want. To do this, we connect through Miro and create a dashboard to facilitate these explanations so that what is most important is recorded.
The result of this session is what I use as a basis for the approach of the project in question virtual office.
User: Miro's navigation was the reason why I decided to use it. It is so good that anyone can adapt to the software with a couple of simple directions.
I love that the templates can be modified with a few clicks to suit the moment's needs.
And their visuals are lovely, complete and easy to understand.

User: Miro has no fundamental weaknesses. It is software that does what it is supposed to do. If it is necessary to use other apps, it is because Miro is not designed for it.

GT:
Users appreciate Figma's remote collaboration capabilities, Jira card integration, canvas design, and various features such as templates, app integrations, presentation tools, and AI components. However, they experience issues when sharing boards with free license users who need to actively collaborate.

Prediction:
User appreciates Miro for its ease of use and ease of use, especially for creating dashboard

2. **Text**:

User: Creating team structures
User: I was able to hop in and create a team structure in less than 5 minutes with no training. I can't remember a tool this simple.
User: I wish it used AI to tell me what I could improve about the layout of my figjam.

GT:
Users appreciate Miro for consolidating scattered information and providing a single source of reference for their team. They find it easy to navigate and enjoy the table features, sticky notes, and embedded comments. However, they face challenges when updating table functions, as using the back button erases all items entered, making editing difficult without losing changes

Prediction:
User finds the tool simple and easy to use, but wishes it had AI capabilities to help improve

3. **Text:**

User: It replaced physical whiteboards for discovery and brainstorming sessions. It also works as an excellent ice-breaker and teambuilding tool to have everyone active during meetings or learning sessions.
User: For those who miss physical whiteboards and post-its this is the solution to easily (and virtually) visualize projects and organize brainstorming, project management and workflow visualization. It's very friendly, intuitive and compatible with other tools.
User: Some toolboxes usually overlap important information, and sharing options are not that easy to understand and use, mainly with external (not frequent) users.

GT:
Users find Miro to be an effective collaboration tool that helps run meetings smoothly, reduces manual work, and saves costs. They appreciate its user-friendly interface, available templates, sharing capabilities, and affordability. Miro has positively impacted their workplace, and they have no significant dislikes about the tool.

Prediction:
User finds the product helpful for brainstorming sessions, ice-breakers, and teambuilding

4. **Text**:

User: It allows me to facilitate remote workshops and keep track of the ideas shared.
User: A lot of features that enable collaborative sessions.
User: Difficulty in organizing the space e find items previously created.

GT:
Users appreciate Miro as their top choice for collaboration with clients, as it allows real-time feedback, process documentation, and easy onboarding. It is particularly useful for remote teams and different time zones. However, users face issues with size capacity, slow performance with too many screenshots, messy boards, and minor formatting problems.

Prediction:
User appreciates the product for its ease of use and collaboration features, but finds it difficult to

5. **Text**:

User: Miro helps me as a Design thinking expert and UX Designer to moderate workshops, collect information, synthesize and visualize insights, do creative work and create the first concepts. And all of this in one place and available to the whole team for us all to work collaboratively.
User: That it is super easy to use, team members can be onbarded in no time and can edit and create content without installing anything. Also the great amount of Templates from the community help a lot.
User: I haven't found anything that bothers me, for what I use miro (collaborative work, moderation for workshops) it meets all my expectations

GT:
Users appreciate Figma's collaboration features, such as commenting and version history, and its design tools, including animation capabilities. However, they find it expensive for larger teams and lacking in advanced design features compared to other software products.

Prediction:
User finds Miro to be a great tool for design thinking and UX designers, as

6. **Text**:

User: Helps quickly get ideas out and keep everyone on the same page.

User: FigJam is intuitive and easy to use, even for non-designers. It's great to use with clients or internal teams to run brainstorms and retros. It has great features like stickers, timers etc that make creating collaborative sessions simple.
User: FigJam can be hard to customize to look clean and polished – it's not necessarily the core use case, it's meant to be casual, but it can sometimes be frustrating when trying to use it for more professional meetings or touchpoints.

GT:
Users appreciate Miro for its collaboration and visibility features, allowing them to store various formats of work and share them with stakeholders or teammates. However, they face challenges in managing multiple open Miro tabs and wish for a better organization system or quick access to specific boards.

Prediction:
User finds Figma to be a great tool for brainstorming and collaboration, especially for design

FUTURE IMPROVEMENTS

1. Manually review Text-Summary pairs having low similarity score after ANN search. Filter out some samples from train data whose Text-Summary seem unrelated/inappropriate.
2. Improve Figma + G2 Summaries
3. Create a pipeline to change the language of Text to English. Eg:

```
User: iPhoneとiPadだと日本語でタイトル入力すると消える！
```

4. Standardize the structure of most frequently occuring Summaries:
5. Include the STRICT_LINKs and try to infer information from it. Multimodal approach. Difficult.
6. Can train larger LLMs like DollyV2, Falcon by making this task as the next token prediction task for the decoder by combining the Text and Summary columns. But it would be difficult to deploy them.

API

Hit a **POST** request with the following configuration

Request body

```
{
 "text": "The text goes here",
"product_name": "Product name goes here",
 "record_type": "Record type goes here"
}

```

API Endpoint: http://52.41.176.142:8000/summarize

Response body

```
{
   "summary": "AI generated summary comes here"
}
```

Sample Python Code for Request

```
import requests
import json

url = "http://52.41.176.142:8000/summarize"

payload = json.dumps({
 "text": "The text goes here",
 "product_name": "Product name goes here",
 "record_type": "Record type goes here"
})
headers = {
 'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```