Tourist Destination Introduction Generation Language Model from Tourist Reviews based on Fine-tuned Flan-T5
SI630 Project Report

Lechen Zhang
leczhang@umich.edu

Abstract

The project builds a generative language model which can automatically generate the introduction of tourist destinations based on the reviews from tourists. The model uses the sentence transformer and TextRank algorithm to find and pick up important sentences, and then used fine-tuned Flan-T5 model with appropriate prompts to encode and decode into an objective, highly generalized introduction. The research verified that TextRank is effective in extracting representative information from complicated data. It also verified that appropriate prompts can significantly improve the performance of seq2seq pretrained model. Our final model in this project research has 37.537 Rouge1 score, which is 240 times higher than the baseline. It verified that pretrained seq2seq models are feasible and promising for complicated text generation tasks.

1 Introduction

The goal of this project is to automatically generate the introduction of tourist destinations based on the reviews of tourists. With the accelerated pace of life, people can rarely spend enough time on planning their vacation trip according to these lengthy tourist reviews. However, the short introductions of tourist destinations are usually provided by the destination itself, which is usually outdated, biased and unappealing. If we could automatically generate the introduction based on the reviews, our tourists can get real-time, authentic and attractive information, while the tourist destinations can also be free from elaborating these introductions.

In this project, I used seq2seq pre-trained deep learning models to generate a general introduction for each tourist destination by summarizing, extracting and restructuring information from plentiful tourist reviews. More specifically, I used the sentence transformer and TextRank algorithm to find important sentences, and then used fine-tuned Flan-T5 model with appropriate prompts to encode these sentences and decode into an objective, highly generalized introduction for tourist destinations.

Previous works on summarizing tourist reviews and generating introductions are usually using traditional and old NLP methods like extracting the frequent words, finding certain information for each section (location, time, environment, etc.), or just using BERT encoder to find keywords. The generated text from these methods are usually too short, subjective and unprofessional.

However, thanks to the rapid development of pretrained seq2seq model like Flan-T5, my project used these newly-built models to make a significant progress on generating a professional, objective, generalized introduction from tourist reviews. The Rouge score of my model reaches 37.537, which is 240 times higher than the baseline of simply picking high-frequency sentences from the review. It verified that pretrained seq2seq models are feasible and promising for complicated text extraction/generation tasks.

Additionally, my project has proved that TextRank is an effective method to extract representative information from complicated data like tourist reviews. My project also proved that appropriate prompts can make a huge difference to the performance of seq2seq pretrained model.

Furthermore, my project revealed a serious problem of fine-tuning seq2seq model that is worth looking into. I found that the low quality of training data can sometimes lead to overfitting problems and make our models to tell lies. If the information between input and output is not one-to-one correspondent (e.g., some histories are missing in reviews but exist in introductions for training), then our model will begin to generate fake information to fill up the missing history. Adding a restriction to generative model should be an important consideration for future research in this field.

2 Data

The dataset mainly came from my SI650 Course project, but I did some fine tuning to make it accommodate our text summarization tasks. The original crawling steps are shown below:

For this project, I use two websites to crawl the introduction and reviews of tourist attractions.

The introduction data came from the official website https://www.city-data.com/articles/, which contains all tourist attractions in the United States and the corresponding introduction. I used *BeatuifulSoup* and *requests* to crawl all pages from the website.

The comment data were crawled from https://www.tripadvisor.com/, which provides many detailed comments to each tourist attraction. Other than comments, we also crawled the ratings and total number of comments of each tourist destinations to help filtering the data. During this process, we used the $requests$ package of Python to simulate normal user access request, and use *BeatuifulSoup* to get all the comments.

However, we have to build up a mapping between the introduction of tourist attraction to the website of it on TripAdvisor. To do so, I used a Google API (https://customsearch.googleapis.com/customsearch/v1). We can send the attraction name to this API, and it will return the website url of TripAdvisor. After all the steps above, I get a dataset with the example data structure shown in Fig. 1.

doc_no	2	**Name**	Alabama Theatre
Rating	4.5	**Review Num**	1797
Introduction	The Alabama Theatre was established in 1927 by Paramount Studios. It was meant to be a place to showcase the Paramount brand of films. It was used as a movie palace" for 55 years. Since 1987 the Alabama Theatre has been a place for Performing Arts. It will host a number of live events and films throughout the year. etc.		
Review	The show was wonerful and we really enjoyed the music. The fact that we were able to take 5 kids and 3 adults for the price of the 3 adults tickets was fantastic. The only dissappointment was that the show was only the singing and dancing and the gentleman who does the juggling and stuff was not there that night and that there was only one instance of acrobatics in the show. The best singer was Rodney Williams, the only Black performer in the troupe. Slim Chance, the juggler, comedian, and ventriloquist, was the most polished professional. There was a variety of music and the dancers were very good, too. etc.		

Figure 1: Example of data structure in dataset

After crawling the dataset, I filtered it by dropping all rows with missing value, and dropping all tourist destinations with fewer than 300 comments. Also, I dropped all comments that are less than 3.0 ratings, since most of them are complaints that are not helpful to generate introduction.

Finally, we got the final dataset with 2524 tourist destinations. The average word counts of introduction is 381, and the average word counts of comments for each tourist destination is 994. The detailed word-count distribution is shown in Fig. 2 and Fig. 3.

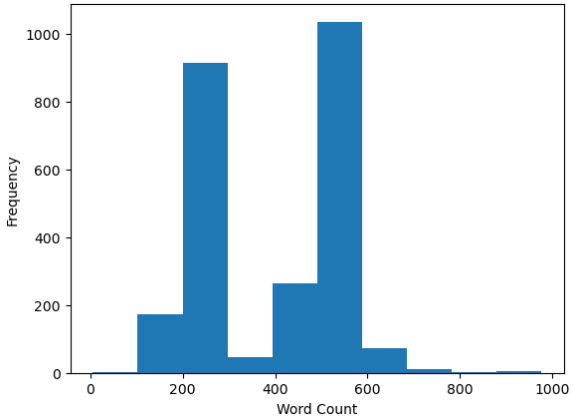

Figure 2: Word count distribution of Introduction

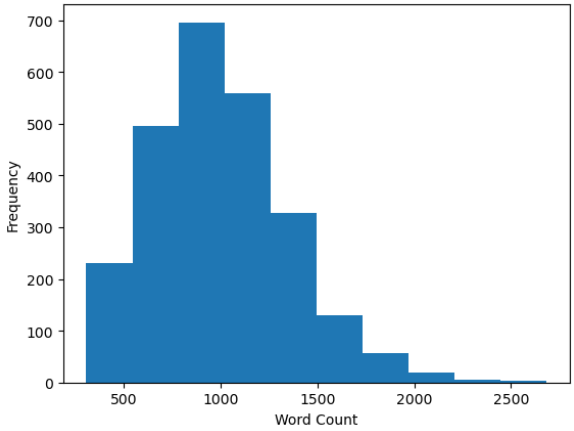

Figure 3: Word count distribution of Reviews

After the project Update, I noticed that the word counts for Reviews are too long for common deep learning tokenizers (994 words on average v.s. 512 tokens). However, instead of simply truncating the reviews, I applied a new invented method called TextRank to pick up important sentences from the Reviews, which will be introduced latter.

3 Related Work

Most research papers about tourist destinations I could find online mainly focus on the basic statistics of travel review texts. Some researchers like

Guerrero-Rodriguez used NLP methods to analyze the text data, but their research team used traditional NLP methods instead of neural networks. The method Guerrero-Rodriguez used is to calculate the "Mutual information" of different reviews, and extract the shared words and phrases. After the extraction, the researchers uses "Jaccard coefficient" to generate representative phrases for each kind of destinations (Guerrero-Rodriguez et al., 2023). The method Guerrero-Rodriguez used is probably too simple for my project, but their research proved the power of information behind these reviews, and reminded me to do primary categorizing before training my neural network.

Tsai's research also focus on using traditional NLP analyze methods to summarize sentences based on the reviews of hotels and resorts. They also use supervised machine learning to categorize reviews, then generate sentences for each category and combine them into a whole paragraph. This might be useful for our travel destination summary, since some information are indispensable for each place (e.g. the transport, location, service, environment). For each review, their research will conduct subjectivity analysis by evaluating the sentence polarity. This might be useful to confirm that our generated introduction is objective and fair. However, Tsai's research is still focusing on extracting feature words and sentences from the original text, which means the generated summary is very likely to be subjective. (Tsai et al., 2020)

Though we could hardly find previous research on objective text generation based on tourist reviews, we can take other Text Summarization research as a reference. There are many research focusing on generating the abstract of article based on its content. For example, Kieuvongngam's research uses BERT and GPT-2 to generate Text Summarization of COVID-19 Medical Research Articles. Though it focused on different topic, we can still reference the evaluation metrics they use, which is ROUGE Score. These metrics can effectively compare the similarity between generated text and target text (Kieuvongngam et al., 2020).

El-Kassas's research team published a comprehensive survey on text summarization methods, which reviewed detailed methodologies, evaluation metrics, possible challenges, and etc. Since our reviews come from different people, it should be classified as multi-document task. According to El-Kassas's recommendation, clustering the documents and pick the most important sentences can help improving the quality of summary. Also, the paper introduced a deep-learning method by using LSTM to encode the documents and then decode it, which is a suitable method for cases when documents are not necessarily the same structure as the summary (since we are using reviews) (El-Kassas et al., 2021).

Liu's paper gives a more specific implementation of deep learning text summarization. They used a encoder-decoder structure, but they proposed a new encoder called BERTSUM, which solved the problem that the original BERT can't handle multi-sentential inputs. They used a 6-layered Transformer for decoder. To solve the mismatch problem between encoder the decoder, they used different optimizers for them during the fine-tune stage (Liu and Lapata, 2019).

In my final model design, I utilized previous idea about Abstractive summarization and invented a new method TextRank to reach the same effect. The idea is just picking up representative sentences from the input data, but it's extremely helpful for complicated input data like user's reviews (which contains a lot of repeated and useless information). Additionally, I've utilized the idea of encoder-decoder structure as a solution to the text summarization task, but instead of training them separately, I used a newly-invented seq2seq model Flan-T5 to do the work, which is more effective and efficient than the previous methods.

4 Methodology

Broadly speaking, I used a seq2seq pretrained language model (Flan-T5-Base) to generate the introduction of tourist destination. However, due to special characteristics of this task, I did some extra work to make this seq2seq model best fit our task.

4.1 TextRank

First of all, since our crawled reviews have 994 word counts on average, which is much higher than the maximum token numbers of T5 model (512 tokens). If we just truncate them at 512 postion, we are about to lose half of the information, which is quite unacceptable. Consider that there are many repeated information in tourists' reviews, I'll first apply a TextRank algorithm with a sentence transformer to extract important sentences before using the Flan-T5 model.

More Specifically, I split all reviews

of a specific tourist destination into sentences (around 80 sentences), and then used a pre-trained sentence transformer `sentence-transformers/all-MiniLM-L6-v2` on HuggingFace (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2), which can transform a sentence into 384 dimensional vector. Then I will use cosine similarity to calculate the similarity matrix of all sentences of this place (around 80x80). After that, I applied a python library `networkx.pagarank` to calculate the PageRank of all these sentences, in order to find the most representative sentences among all sentences without repetition. Finally, I added the top ranked sentences into my final dataset for training and testing.

The idea partially came from Prateek Joshi's Blog (Joshi, 2018), but instead of using traditional GloVe encoder, I used the sentence transformer instead, which can provide a far better sentence encoding. However, the idea of "TextRank" in this article is inspirational to me.

4.2 Flan-T5 model and Prompt Selection

Basically, I will use the pretrained seq2seq model on HugglingFace (https://huggingface.co/), which is `google/flan-t5-base` (https://huggingface.co/google/flan-t5-base). It accepts the text input and outputs the text output, which best fits our text generation task. By adding simple prompts to the input text, the model can finish all jobs for you, and the most simple one is just adding "summarize:" to the beginning.

Also, since our input information is quite limited (it's almost impossible to restore a perfect introduction based on the tourist reviews), it's very likely that our output text starts repeating itself or stop very early. Therefore, I added 2 restrictions: Firstly, the output token size should between 100 and 300. Secondly, there shouldn't be any repeated 3-gram in the output text. These two restrictions help us generate a introduction that has appropriate length and structure.

However, since we are generating tourist destination introduction instead of a simple summarization task, the quality of our T5 model prompt matters. If we simply use "summarize:" as prompt, the generated text will be very subjective like "I like this place". Also, since the name of tourist destination is quite important (it usually occur at the beginning of each introduction, but rarely can travellers mention it), it might be helpful to add the NAME field into prompt. Therefore, it's worth testing different prompts and pick the best performing one for this task. In this project, I'm going to test 4 different prompts on untrained Flan-T5 model, which are listed below:

- **Prompt1**: "summarize:" + TEXT
- **Prompt2**: "Paraphrase from an objective perspective:" + TEXT
- **Prompt3**: "Write an introduction from reviews:" + TEXT
- **Prompt4**: "Write an introduction from reviews about" + NAME + ":" + TEXT

4.3 Flan-T5 Model Fine-tuning

Other than prompt selection, model fine-tuning is also very helpful for us to customize model for specific tasks, especially for generating the introduction of tourist destination. Since we are converting subjective words into objective introduction, it's very likely that the model still keeps a subjective perspective. Also, it's important for our model to learn the traditional structure of tourist destination introduction (e.g., xx is a historical museum located at xxx), thus improve the overall performance.

The dataset is split into $80\% : 20\%$ for training and testing. The training is based on the Introduction field in my dataset. Its average length is 381, which means we can safely truncate it at 512 token size. Fortunately, Flan-T5 model provides the built-in loss function for seq2seq model, so I only need to set up all parameters and evaluation metric before training. The important train parameters are shown below:

- Total Epoch: 3
- Batch size: 4
- Learning Rate: 4e-5
- Best model metric: Rouge1

5 Evaluation and Results

5.1 Evaluation Metrics

I will mainly use ROUGE Score and human evaluation to evaluate quality of summarization. More specifically, I will use the F1 score of ROUGE-1, ROUGE-2, ROUGE-L, and 5-score human evaluation as my evaluation metrics.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score measures the overlap between generated and reference text, which is the most common evaluation method in text summarization. It compares the shared n-grams (contiguous sequences of words) between the two summaries. ROUGE-1 refers to the overlap of unigram, ROUGE-2 refers to the overlap of bigrams, and ROUGE-L refers to the "Longest Common Subsequence (LCS)". The ROUGE score will return "Precision", "Recall", and "F-score", and we use "F-score" here to evaluate both "Precision" and "Recall" at the same time.

As for human evaluation, the 5-scale evaluation metric is shown below:

1. The summary is not readable.
2. The summary is partially readable, but the contents can't reflect the fact of reviews.
3. The summary is mostly readable, and some contents are related to the reviews.
4. The summary is readable, and the contents are related to the reviews. But the summary is not objective enough as an introduction.
5. The summary is readable, highly summarized, objective, and captured all features of the tourist destination.

5.2 Baseline Model

The first baseline I will use is the randomly generated paragraph based on the words in reviews. The second baseline I will use is automatically picking up the sentences that share the most popular words of the destination, and combine them into a paragraph.

The evaluation result of baselines are shown in Fig. 4. As we can see from the result, the second baseline outperformed baseline 1 in all evaluation metrics, especially in Rouge-2 and Human evaluation. The main reason is that the second baseline picked complete sentences as its output, which is very likely to share bigrams with the ground truth. However, the first baseline only randomly pick up words, which means it's very unlikely to have readable phrases.

5.3 TextRank Performance in Flan-T5

I tested the model's performance with/without TextRank on untuned Flan-T5 model, which is shown in Table. 1. Since we need prompts to make Flan-T5 work, we chose Prompt4 for comparison.

	Model	Rouge1	Rouge2	RougeL	Human Eval
0	Baseline1	0.106740	0.004203	0.068473	1.10412
1	Baseline2	0.155052	0.023883	0.091625	1.84242

Figure 4: Evaluation Results for Baselines

Model	Rouge1	Rouge2	RougeL	Human Evaluation
Baseline1	0.1067	0.0042	0.0685	1.1041
Baseline2	0.1551	0.0239	0.0916	1.8424
Prompt4 without TextRank	24.0364	4.6408	13.8634	3.345
Prompt4 with TextRank	24.9597	5.1283	14.3126	3.41

Table 1: Comparison Table of TextRank

Also, I included an example that compares the beginning 3 sentences of reviews with/without TextRank, which is shown below:

Without TextRank:

- Must see in Birmingham!.
- Do not miss this tour!
- A fantastic tour!.

With TextRank:

- We toured this church steeped in civil rights history last week and I must admit the church members did a fantastic job making sure every got into the tour to see all of the artifacts and understand the history.
- We also had the honor of meeting and chatting with the pastor of the church (since 2002) Reverend Arthur Price, Jr. Our visit was a very educational and moving experience.
- Even though I couldn't go inside, it was very important to me to pay homage to the brave people who used this church as the base for civil rights activity and to honor the four young girls killed there by a Ku Klux Klan bomb.

5.4 Prompt Comparison

I tested the model's performance of all listed prompts on untuned Flan-T5 model, which is shown in Table. 2.

Also, I included an example that compares the output text of Prompt1 and Prompt4 about **Pinball Hall of Fame**, which is shown below:

- **Prompt1** ("summarize:" + TEXT): Great place to visit. Cheap to play games which

Model	Rouge1	Rouge2	RougeL	Human Evaluation
Baseline1	0.1067	0.0042	0.0685	1.1041
Baseline2	0.1551	0.0239	0.0916	1.8424
Prompt1	19.6194	3.4308	11.3094	2.714
Prompt2	21.8264	4.1061	12.4791	2.853
Prompt3	24.3739	4.633	13.6929	3.329
Prompt4	24.9597	5.1283	14.3126	3.41

Table 2: Comparison Table of different Prompts

was needed after being down the strip. Lots of pinball machines to choose from, anything that you can imagine. Got our picture taken at “Welcome to Las Vegas” sign as well as it’s just across the street and a 5 minute walk from the Pinball building. Many of the pinball games required 50 cents but worth it.

- **Prompt4** ("Write an introduction from reviews about" + NAME + ":" + TEXT): Pinball Hall of Fame is a great place to go with friends and pinball machines are a must. It is surrounded by a huge selection of pinball games and arcade games. The place is huge and clean and has a lot of machines. The arcade games are incredibly good and the staff is friendly. The pinball game selection is huge, and the prices are reasonable. The games are very good and are very well maintained. The staff is very friendly and helpful. The location is great and the parking is great.

5.5 Flan-T5 Tuning Results

After 3 epochs of training, I got the performance charts of loss on train dataset in Fig. 5.

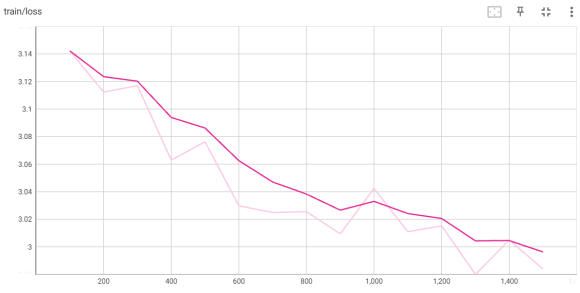

Figure 5: Loss in train dataset

Furthermore, I got the ROGUE score (Rouge1, Rouge2, RougeL) charts on evaluation dataset in Fig. 6.

Figure 6: ROGUE score in Evaluation dataset

The best performance model during training is shown in Table. 3

Model	Rouge1	Rouge2	RougeL	Human Evaluation
Baseline1	0.1067	0.0042	0.0685	1.1041
Baseline2	0.1551	0.0239	0.0916	1.8424
Prompt4 without TextRank	24.0364	4.6408	13.8634	3.345
Prompt4 with TextRank	24.9597	5.1283	14.3126	3.41
Prompt 4 with TextRank and Fine-tuning	37.5372	10.7559	20.609	4.535

Table 3: Comparison Table of Fine-Tuning

Here’s an example of generated text for **Pinball Hall of Fame**:

`The Pinball Hall of Fame is located in Las Vegas, Nevada. The Pin Ball Hall of fame is a museum dedicated to pinball. The museum is open from 10am to 5pm daily. The pinball hall of fame was founded in 1939 by the late Countess of Las Vegas. The hall of Fame was built in 1931 and is located at 10801 Las Vegas Boulevard. The building was built by the Las Vegas Board of Supervisors. The Hall of the Fame is open to the public from 10 am to 5 pm daily. There are a number of pinball machines and arcade games available. The arcade games`

are based on the arcade games of the 1980s. There is sandboxes, arcades, and arcades. Several of the pinball games are available for purchase. The main attraction is the Pinball Museum. The Museum is located on the Las Las Vegas Strip. The exhibit is based in the Pin Ball Museum. There were a total of 108 pinball machine and arcade machines. The center of the museum is located near the Las vegas strip. The Center of the Pinbow Hall of Excellence is located next to the Las Caesars Palace.

6 Discussion

6.1 TextRank

Based on the results about TextRank in Table. 1, we can find that it helps improving both the Rouge score performance and the Human Evaluation score. This improvement can also be verified by the example. Without TextRank, the sentences are very likely to have same meaning (e.g., must see and don't miss the tour). Also, the sentences are usually too short that hardly contain useful information. However, with TextRank, the top-ranked sentences usually contain rich and representative information with almost no repetition.

Therefore, we can conclude that TextRank is helpful in extracting the core information, especially for complicated data like reviews from tourists. The precious token length limit should be used for more representative information.

6.2 Prompt Picking

Based on Table. 2, we can find that the quality of prompts can have a huge impact on the model performance, especially for complicated tasks. Since we are writing introduction rather than simple summarization, the prompt should be detailed enough so that the model can understand that we want a rephrased expression. This can be inferred by comparing the ROUGE score for **Prompt1** with **Prompt2-4**. Furthermore, by comparing **Prompt3** and **Prompt4** we can found that adding the name of tourist destination to the prompt can also bring significant improvement to the model performance. The reason is that NAME usually occur at the beginning of each introduction, and it's helpful to emphasize it in the prompt in order to make our model realize its importance.

This improvement can also be verified by the example in Result section. From the example we can see that the outputs of **Prompt1**("summarize" + TEXT) are just the short version of each tourist review, like "great place to visit", "cheap place", etc., which is not the introduction we want. However, the performance of **Prompt4** ("Write an introduction from reviews about" + NAME + ":" + TEXT) is far better than the previous one. It started from "Pinball Hall of Fame is a great place ...", which is a standard beginning of tourist destination introduction. Furthermore, it high summarized the characteristics of the place from multiple aspects in an objective tone (Huge, Clean, Lot of machines, good arcade games, friendly staff, etc), which is quite professional. Though the vocabulary is a bit limited (many "very" and other simple words), we can still conclude that it's a very good introduction for tourist destination.

6.3 Flan-T5 Fine-tuning

In general, the fine-tuning significantly increased the performance of our model, which can be seen from the Table. 3. The fine-tuned model has 37.5372 Rouge1 score, which is much higher than the 24.9597 score for untuned model. Furthermore, the Rouge2, RougeL and Human Evaluation score is also much higher in tuned model. This indicates that we got a sucess in tuning the Flan-T5 model for this task.

Also, our success can be verified by the example for **Pinball Hall of Fame**. As we can see from the example, the generated text is quite professional that I can't even tell whether it's generated by model or came from ground truth data. It started from the location of the place, and then the opening time, and then the history of the place, and then the characteristics of the place, which is amazing.

As for the training process, we can see from Fig. 5 and Fig. 6 that the loss is keep reducing during the training process, and the ROUGE1 score keeps increasing. However, the ROUGE2 and ROUGEL score is not very stable. The ROUGE2 score follows a slow increasing trend, but the ROUGEL score just oscillate around 20.5.

Additionally, there's a hidden problem that the generated information is not necessarily right. From the example mentioned above, we can know that the the hall of Fame is located at 10801 Las Vegas Boulevard. However, it's partially true. It's true that the hall is located at "Las Vegas Boulevard", but 10801 is a fake number which doesn't exist in real world. Furthermore, according to Wikipedia,

the hall was built in 2006, which is different from Year 1931 in the generated text. It's quite frustrating that our model is lying to improve its ROUGE score.

The possible reason is that the ground truth (Introduction filed) in my training/testing data is not good enough. When I look at the data, I found that many sentences are almost impossible to be generated based on tourists' review, e.g., the history of the place, the opening hours, the interesting stories, etc. Also, some information in tourist reviews don't have corresponding sentences in true data, e.g., the friendly service, the environment, etc. This inconsistency may cause our model to be confused when generating N-grams. But since 1-grams are usually only related to the characteristics of the place, it makes sense that ROUGE-1 score is keep increasing. However, the wrong N-grams are usually leading to factual errors in our generated text.

7 Conclusion

In this project, I used the sentence transformer and TextRank algorithm to find and pick up important sentences, and then used fine-tuned Flan-T5 model with appropriate prompts to encode these sentences and decode into an objective, highly generalized introduction for tourist destinations.

During the research, I've verified that TextRank is an effective method to extract representative information from complicated data like tourist reviews. My project also proved that appropriate prompts can make a huge difference to the performance of seq2seq pretrained model, and it's important to provide prompts with detailed and important information for the task. Finally, my project verified that pretrained seq2seq models are feasible and promising for complicated text extraction/generation tasks. The Rouge score of my model reaches 37.537, which is 240 times higher than the baseline of simply picking high-frequency sentences from the review.

Furthermore, my project revealed a possible future research direction of fine-tuning seq2seq model. The low quality of training data can sometimes lead to overfitting problems and make our models to tell lies. Therefore, future researchers may need to add more restrictions to generative model to prevent it from outputting fake information.

Github Link:

https://github.com/orange0629/si630proj

8 Other Things We Tried

In general, I think most things I tried in this project are quite successful, which is probably because Flan-T5 is too powerful itself. But during the very early stage when I didn't know about T5 model, I met the following difficulties, and some methods I've tried didn't work well.

Initially, I tried to use BERT encoder to encode the comments and extract keywords, then use LSTM to generate introductions from the word vectors. However, the performance is bad since LSTM is not a pretrained model. Since we don't have too many training data on hand, it's almost impossible to train a generative model to generate readable sentences through this method.

After that, I tried to use GPT2 decoder to decode the keywords into complete sentences. However, it turned out that it's too hard to tune the encoder and decoder separately, since I'm still using BERT to encode and find the keywords in tourist reviews. Also, the performance is much worse than the Flan-T5 model without fine-tuning because I'm losing too much information during the keyword extraction. Therefore, I chose to use Flan-T5 as our last model, because it can both encode and decode by itself, which is much easier to tune. Since Flan-T5 was trained with prompts initially, it's also much more user-friendly method to finish our generative work.

9 What You Would Have Done Differently or Next

In general, I think my project is quite successful in solving the Tourist Destination Introduction Generation problem. However, the problems I met during my project are still worth noticing and needs extra future work.

First of all, the quality of training data should be reconsidered. In my project I used the ground truth data of Introduction from a government travel website, which is professional but too formal. It contains the history of the tourist destination, as well as the related story, celebrity or culture, which is almost impossible to be inferred from tourist's review (you can't expect tourists to know so many history stories). It can make our model be confused since they have no idea where these sentences came

from, which caused the generated introduction having wrong historical stories frequently.

Furthermore, we should do some work to ensure our model doesn't tell a lie. It's unethical and even harmful to provide potential tourists with wrong information, such as wrong history, wrong location, wrong opening time, etc. However, during the fine-tuning the model usually learn to keep the same output structure as the train data, which usually means the model will make up a fake story at the beginning even if the model didn't get it from the reviews. In the future work, other than improving the train data quality, I think I should probably add heavier punishment to "wrong information" and inaccuracies during the fine-tuning state. Also, we may need to implement human oversight to manually select from different tuned model, in order to pick up the model that doesn't tell a lie.

Also, due to the time limit I haven't tried to train more epochs during the fine-tuning (currently 3 epochs). According to the training charts for ROUGE1 score and Loss value, it still haven't converged yet, so what will happen to the performance if the model fully converge? I'm quite curious about how epoch number will interfere with the model performance. Will it make our model be better at telling a lie (overfitting?), or will it make our model tell more accurate information (better fitting?). I believe it can be an interesting future topic.

References

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Rafael Guerrero-Rodriguez, Miguel Á. Álvarez Carmona, Ramón Aranda, and Adrián Pastor López-Monroy. 2023. Studying online travel reviews related to tourist attractions using nlp methods: the case of guanajuato, mexico. *Current Issues in Tourism*, 26(2):289–304.

Prateek Joshi. 2018. An introduction to text summarization using the textrank algorithm. *Analytics Vidhya*.

Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. Automatic text summarization of covid-19 medical research articles using bert and gpt-2. *arXiv preprint arXiv:2006.01997*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders.

Chih-Fong Tsai, Kuanchin Chen, Ya-Han Hu, and Wei-Kai Chen. 2020. Improving text summarization of online hotel reviews with review helpfulness and sentiment. *Tourism Management*, 80:104122.