ĐẠI HỌC QUỐC GIA, THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Xử Lý Ngôn Ngữ Tự Nhiên (CO 3085)

Text Summarization

GVHD: Võ Thanh Hùng
SV thực hiện: 2211612

TP Hồ Chí Minh, Tháng 10 Năm 2024

Mục lục

1 Ý tưởng thực hiện

Đề tài này lấy ý tưởng tóm tắt một bài báo trên trang web cnn.com

1.1 Chọn phương thức tóm tắt

1.1.1 Extractive summarization

Extractive summarization là một cách thức tóm lược văn bản dựa trên những câu tồn tại trong bài, hay nói cách khác là lấy những câu quan trọng làm tóm lược.
Extractive summarization gồm 4 giai đoạn:

1. Tạo biểu diễn trung gian
2. Đánh giá câu
3. Chọn câu
4. Ghép các câu đã chọn để tạo tóm lược

Thực thi Extractive summarization:

1. Neural network: BERT
2. Graph-based: TextRank

Tuy có thể tóm lược văn bản nhưng phương thức này gặp vấn đề trong việc sinh ra một tóm lược không lưu loát và bị trùng lặp thông tin.

1.1.2 Abstractive summarization

Abstractive summarization tóm lược văn bản thông qua việc tạo câu mới dựa trên ý nghĩa của đoạn văn hơn là trích câu quan trọng, từ khóa.
Thực thi Abstractive summarization:

- Neural network: BART, T5, PEGASUS, ...

Với đề tài này, ta chọn phương thức abstractive summarization.

1.2 Chọn model

Vì đề tài này dựa trên việc tóm lược bài báo CNN, ta có thể sử dụng các mô hình đã đc fine-tuned như bart-large-cnn của facebook, pegasus-cnn_dailymail của google.
Tuy rằng pegasus tóm lược báo CNN tốt hơn BART nhưng mặt trái là pegasus lại nặng hơn nhiều so với BART.
Với đề tài này, ta sẽ hiện thực với BART để tiết kiệm chi phí, tuy nhiên nếu không màng chi phí có thể hiện thực với pegasus

2 Hiện thực

Link code: https://github.com/Khoawawa/text-summarization/blob/main/text_summarization.ipynb

2.1 Các thư viện

```
!pip install -U transformers # huggingface models library
!pip install -U datasets # huggingface dataset library
!pip install -U accelerate # enhanced calculating library
!pip install -U evaluate # evaluate models library
!pip install -U requests # http request library
!pip install -U bs4 #parsing HTML library
!pip install -U bert-score #bert score library
!pip install -U torch # pytorch for device specification
```

2.2 Định dạng máy sử dụng

Nếu như kết nối với GPU thì chương trình sẽ chạy model với GPU, đồng thời vẫn hỗ trợ sử dụng CPU.

```
import torch
device = "cuda:0" if torch.cuda.is_available() else "cpu"
```

2.3 Tải BART và dataset CNN dailymail

Với dataset, ta chỉ lấy phần test

```
tokenizer = AutoTokenizer.from_pretrained("facebook/bart-large-cnn")
model = AutoModelForSeq2SeqLM.from_pretrained("facebook/bart-large-cnn").to(device)
ds_test = load_dataset("abisee/cnn_dailymail","3.0.0", split = "test")
```

2.4 Hàm tóm lược

Với hàm tóm lược này, ta chỉ tóm lược 1024 tokens (độ dài đầu vô tối đa của BART) đầu của text. Bài báo thường đưa những nội dung quan trọng nhất vào đầu bài nên ta dựa vào tính năng này để tóm lược.
Với các bài tóm lược, độ dài được khuyến cáo tối đa là 250 từ nên ta cài đặt max_length là 250 tokens.

```
def summarize(text,max_length=250,min_length=30):
    tokenized_text = tokenizer(text,
                               max_length = 1024,
                               padding = "max_length",
                               truncation = True,
                               return_tensors = "pt"
                               ).to(device)

    output = model.generate(
        tokenized_text["input_ids"],
        max_length = max_length,
        min_length = min_length
        )
```

```
    summary = tokenizer.decode(output[0], skip_special_tokens=True)

    return summary
```

2.5 Thực thi tóm lược để đánh giá

Tạo 2 list cho tóm lược của bart và tóm lược của dataset. Ta mặc định số bài báo để đánh giá là 100 bài báo.

```
bart_summaries = []
ref_summaries = []
no_eval_articles = 100

for i in range(no_eval_articles):
    article = ds_test[i]['article']
    summary = ds_test[i]['highlights']
    # SUMMARIZE
    bart_summary = summarize(article)

    bart_summaries.append(bart_summary)
    ref_summaries.append(summary)
```

2.6 Đánh giá

Sử dụng bert score để đánh giá dựa trên 3 phương thức f1, precision và recall.

```
from evaluate import load

bert_score = load("bertscore")

results = bert_score.compute(predictions=bart_summaries, references=ref_summaries, model_type="facebook/ba
score = {
    'f1': results['f1'],
    'precision':results['precision'],
    'recall': results['recall']
}
import numpy

print(f"F1: {numpy.average(score['f1'])}")
print(f"Precisions: {numpy.average(score['precision'])}")
print(f"Recall: {numpy.average(score['recall'])}")
```

2.7 Tạo box plot

```
import matplotlib.pyplot as plt
# Prepare the data for box plot
```

```
data = [results['f1'], results['precision'], results['recall']]
labels = ['F1 Score', 'Precision', 'Recall']

# Create a box plot
plt.figure(figsize=(8, 5))
plt.boxplot(data, labels=labels)

# Adding labels and title
plt.title('Box Plot of Performance Metrics')
plt.ylabel('Scores')
plt.ylim(0, 1)  # Set y-axis limits
plt.grid(axis='y')
plt.show()
```

2.8 Lấy dữ liệu từ trang web

Nếu ta inspect trên cnn.com một bài báo ngẫu nhiên nào đó, ta có thể thấy mọi nội dung của bài báo đều nằm trong

```
import requests
from bs4 import BeautifulSoup

def scrape_cnn_article(url):
    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.content, "html.parser")

        # For CNN articles
        if "cnn.com" in url:
            title = soup.find('h1').get_text()
            article_body = soup.find_all('p', class_="paragraph inline-placeholder vossi-paragraph")
            content = " ".join([p.get_text() for p in article_body])
            return title, content
        else:
            return None,None

    else:
        print(f"Failed to retrieve the article. Status code: {response.status_code}")
        return None, None
```

3 Đánh giá

Trong phần đánh giá này, ta sử dụng Bert score để đánh giá model.
Bert score có 3 tiêu chí đánh giá:

1. precision: chênh lệch giữa phần tóm lược sinh ra bởi model với phần tóm lược chuẩn của bài báo

2. recall: model giữ lại được bao nhiêu thông tin quan trọng
3. F1: đánh giá tổng thể dựa trên precision và recall

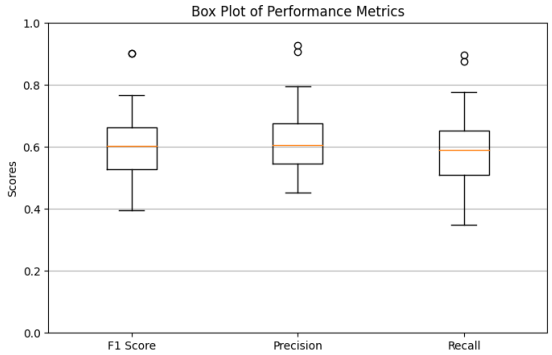

Figure 1: Box plot của F1, precision và recall

Ta có thể thấy với BART ta đảm bảo ít nhất 50% precision và recall. Tuy nhiên trung bình thì chỉ đạt được 62% cho precision và 59% cho recall với tổng thể là 60% cho F1. Nhìn chung, model hoạt động ở mức tầm trung. Tuy đảm bảo yêu cầu nhưng vẫn còn có thể cải tiến.

4 Mô phỏng

Website: https://edition.cnn.com/2024/10/20/politics/mcdonalds-donald-trump-pennsylvania/index.html
Tóm lược: Donald Trump stopped by a McDonald's in Pennsylvania during his Sunday swing. He handed customers food through the drive-thru window, telling them he had made it himself. It's the same job Vice President Kamala Harris has said she held as a young woman. Trump has grown fixated on Harris' employment there.

5 Đề xuất phát triển thêm

- Kết hợp giữa extractive summarization và abstractive summarization: Với extractive summarization có thể đảm bảo bài tóm lược lấy được những ý chính, quan trọng và qua abstractive summarization có thể khiến bài lưu loát hơn.
- Nếu cấu hình máy tốt, có thể thay tokenizer và model từ BART thành PEGASUS.