Course: Natural Language Processing

FINAL PROJECT

Text Summarization

Bidirectional Auto-Regressive Transformer

Instructor: Prof. Le Anh Cuong

Group:

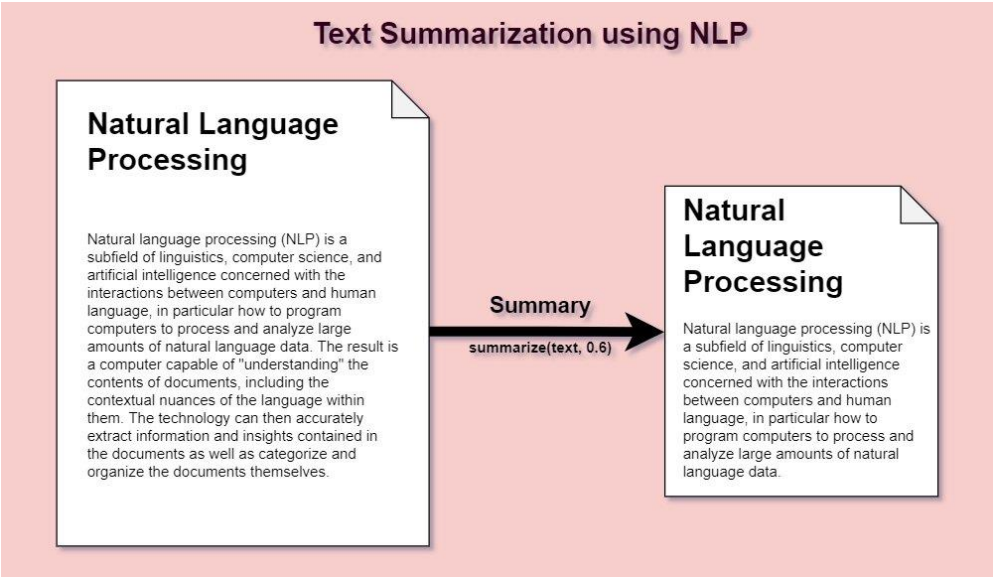
- 520K0127 – Do Pham Quang Hung
- 520K0343 – Le Phuoc Thinh

Outline

I. Introduction
II. BART
III. Experiment conduction result
IV. References

I. Introduction

- Text Summarization is a subtask of Natural Language Processing (NLP) **to generate a short text but contains main ideas of a reference document**. It maybe an impossible mission but thanks to the development of technology, nowadays we can create a model to generate from many texts that convey relevant information to a shorter form.



I. Introduction

There are two main types of text summarization: **extractive and abstractive**.

Extractive summarization involves *selecting and combining the most important sentences or phrases from the original text to create a summary.* This method is simpler and more straightforward but may result in summaries that are not very fluent or readable.

I. Introduction

Extractive summarization example

Original text: According to a new study, *people who eat a plant-based diet rich in fruits, vegetables, whole grains, and nuts may have a lower risk of developing Type 2 diabetes. Researchers found that people who followed a plant-based diet had a 23% lower risk of developing Type 2 diabetes, compared to those who followed a diet that was low in plant-based foods. The study also found that those who consumed more plant-based foods were more likely to have a healthy body weight, which is also linked to a lower risk of diabetes*. However, the researchers noted that more studies are needed to confirm these findings.

Extractive summary: People who eat a plant-based diet rich in fruits, vegetables, whole grains, and nuts may have a lower risk of developing Type 2 diabetes, according to a new study. Researchers found that people who followed a plant-based diet had a 23% lower risk of developing Type 2 diabetes, compared to those who followed a diet that was low in plant-based foods. Those who consumed more plant-based foods were also more likely to have a healthy body weight, which is linked to a lower risk of diabetes.

I. Introduction

Abstractive summarization on the other hand, involves generating new sentences that capture the main ideas and meaning of the original text. This method is more challenging but can produce more fluent and coherent summaries. Abstractive summarization can be further classified as single-document summarization or multi-document summarization, depending on whether the summary is generated from a single document or multiple documents.

I. Introduction

Abstract summarization example

Original text: The United States and China are engaged in a trade war that has resulted in tariffs being imposed on a range of goods, including automobiles, electronics, and agricultural products. The trade war began in 2018 when the US imposed tariffs on steel and aluminum imports from China, and China responded by imposing tariffs on US goods. Since then, both countries have imposed additional tariffs on each other's goods, leading to a decline in trade and economic growth. The trade war has also had global implications, with other countries being affected by the disruption in global trade.

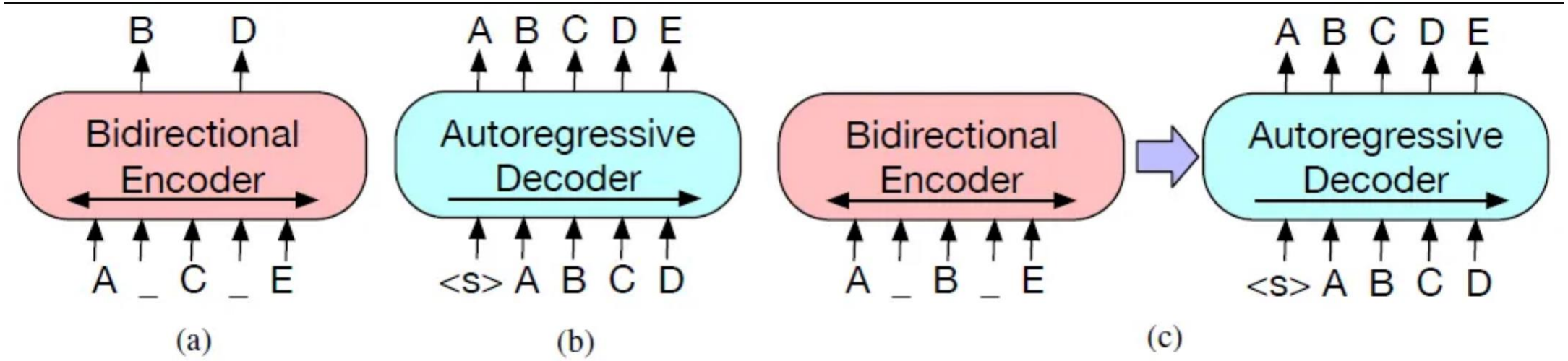
Extractive summary: The United States and China are currently engaged in a trade war that has resulted in tariffs being imposed on a range of goods. The dispute began in 2018 when the US imposed tariffs on steel and aluminum imports from China, which led to China imposing tariffs on US goods. Since then, both countries have imposed additional tariffs, which has caused a decline in trade and economic growth. The global economy has also been affected by this disruption in trade, as other countries have felt the impact of this ongoing conflict.

Outline

I. Introduction
II. BART
III. Experiment conduction result
IV. References

II. Bidirectional Auto-Regressive Transformer (BART)

BART (Bidirectional and Auto-Regressive Transformer) is a language generation model developed by **Facebook AI Research** (FAIR) in 2019. It is a variant of the transformer architecture used in other language models like GPT (Generative Pre-trained Transformer) and uses a bidirectional encoder-decoder architecture to generate high-quality text.



Bidirectional **A**uto-**R**egressive **T**ransformer

- BART is trained on a combination of denoising autoencoder objectives and sequence-to-sequence (seq2seq) objectives, which allows it to handle a variety of natural language tasks, including ***summarization, question answering, and text generation***. It also includes a masking mechanism that allows it to generate missing words in a sentence, which is useful for tasks like language translation.

- One of the key features of BART is its ability to generate both coherent and fluent text, making it well-suited for text generation tasks. It has achieved state-of-the-art results on a number of benchmarks, including summarization and question answering, and is widely used in natural language processing research and applications.

Bidirectional Auto-Regressive Transformer

1.1. Comparison With BERT & GPT

- **(a) BERT**: Random tokens are replaced with masks, and the document is encoded bidirectionally.
- **(b) GPT**: Tokens are predicted auto-regressively, meaning GPT can be used for generation.
- **(c) Proposed BART**: Here, a document has been corrupted by replacing spans of text with mask symbols. The **corrupted document (left)** is **encoded with a bidirectional model**, and then the **likelihood of the original document (right)** is calculated with an **autoregressive decoder**.

Bidirectional Auto-Regressive Transformer

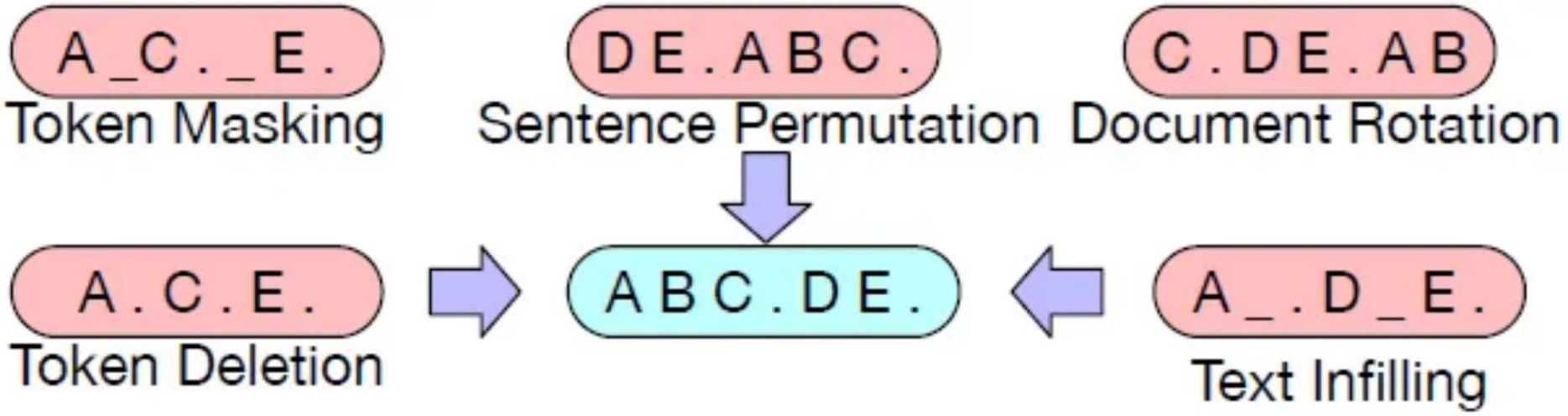
1.2. Architecture

- BART uses the standard sequence-to-sequence Transformer.
- But following GPT, GELU is used instead of ReLU.
- There are **base** and **large** sized BART, which uses use **6** and **12 layers** in the encoder and decoder, respectively.

Bidirectional Auto-Regressive Transformer

1.3. Pretraining

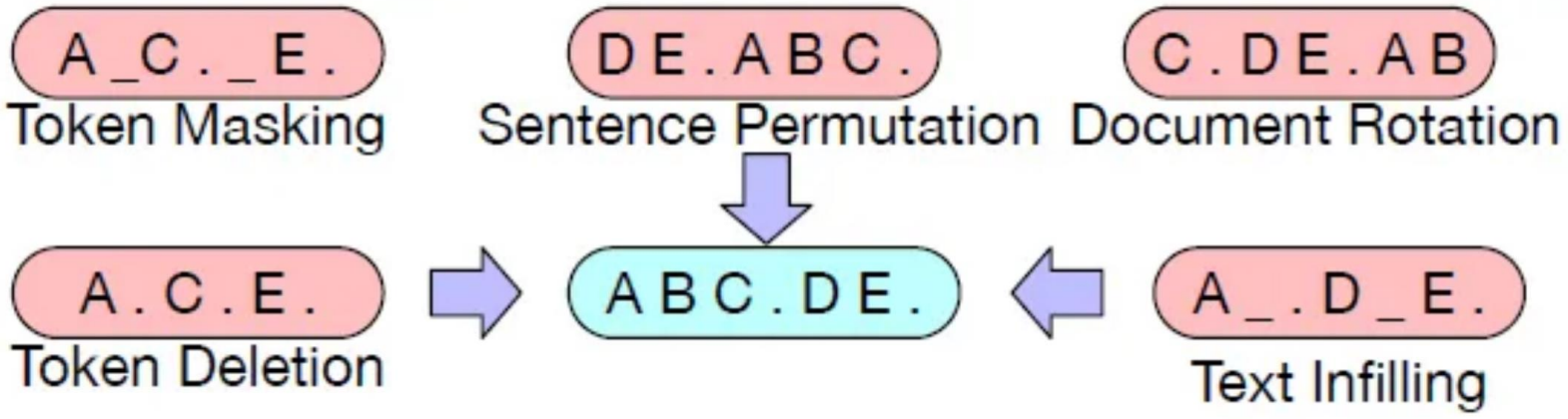
- **Token Masking**: **Same as BERT**, random tokens are sampled and replaced with [MASK] elements.
- **Token Deletion**: Random tokens are deleted from the input. The model must **decide which positions are missing inputs.**



Bidirectional Auto-Regressive Transformer

1.3. Pretraining

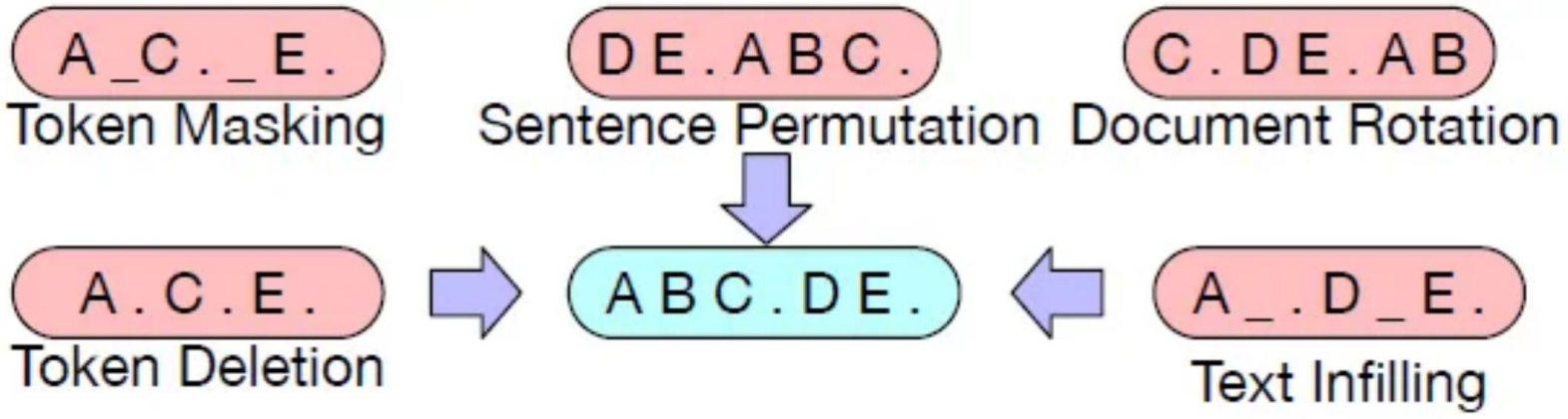
- **Text Infilling**: Text infilling is inspired by SpanBERT where SpanBERT samples span lengths, replaces each span with a sequence of [MASK] tokens of exactly the same length. Text infilling teaches the model to **predict how many tokens are missing** from a span.
- **Sentence Permutation**: A document is divided into sentences based on full stops, and these sentences are **shuffled in a random order**.
- **Document Rotation**: The **document is rotated** so that it begins with that token. This task **trains the model to identify the start of the document.**



Bidirectional Auto-Regressive Transformer

1.3. Pretraining

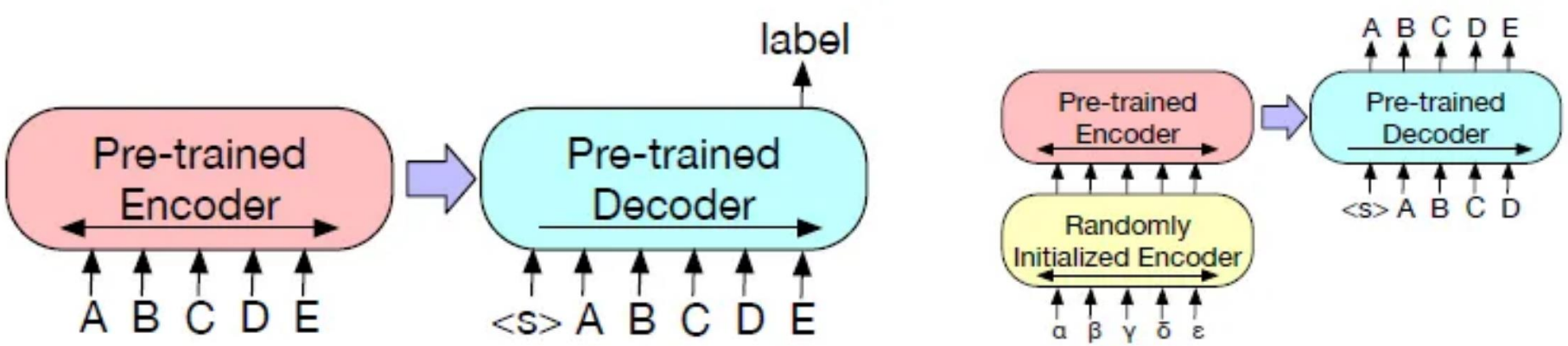
- **Text Infilling**: Text infilling is inspired by SpanBERT where SpanBERT samples span lengths, replaces each span with a sequence of [MASK] tokens of exactly the same length. Text infilling teaches the model to **predict how many tokens are missing** from a span.
- **Sentence Permutation**: A document is divided into sentences based on full stops, and these sentences are **shuffled in a random order**.
- **Document Rotation**: The **document is rotated** so that it begins with that token. This task **trains the model to identify the start of the document.**



Bidirectional Auto-Regressive Transformer

1.4. Fine-Tuning

- **Classification (Left)**: The complete document is fed into the encoder and decoder, and **the top hidden state of the decoder is used as a representation for each word**. This representation is used to **classify the token**.
- **Sequence Generation Tasks**: The encoder input is the input sequence, and the decoder generates outputs autoregressively.
- **Machine Translation (Right)**: BART's encoder embedding layer is replaced with a new randomly initialized encoder. The new encoder can use a separate vocabulary from the original BART model.



Outline

I. Introduction
II. BART
III. Perfomance
IV. Experiment conduction
V. References

3. RESULTS

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

3. RESULTS

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

A combination of text infilling and sentence permutation is used for pretraining large-size BART.

- *Overall, BART performs similarly, with only small differences between the models on most tasks, suggesting that BART's improvements on generation tasks do not come at the expense of classification performance.*

3. RESULTS

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Outline

I. Introduction
II. BART
III. Perfomance
IV. Experiment conduction
V. References

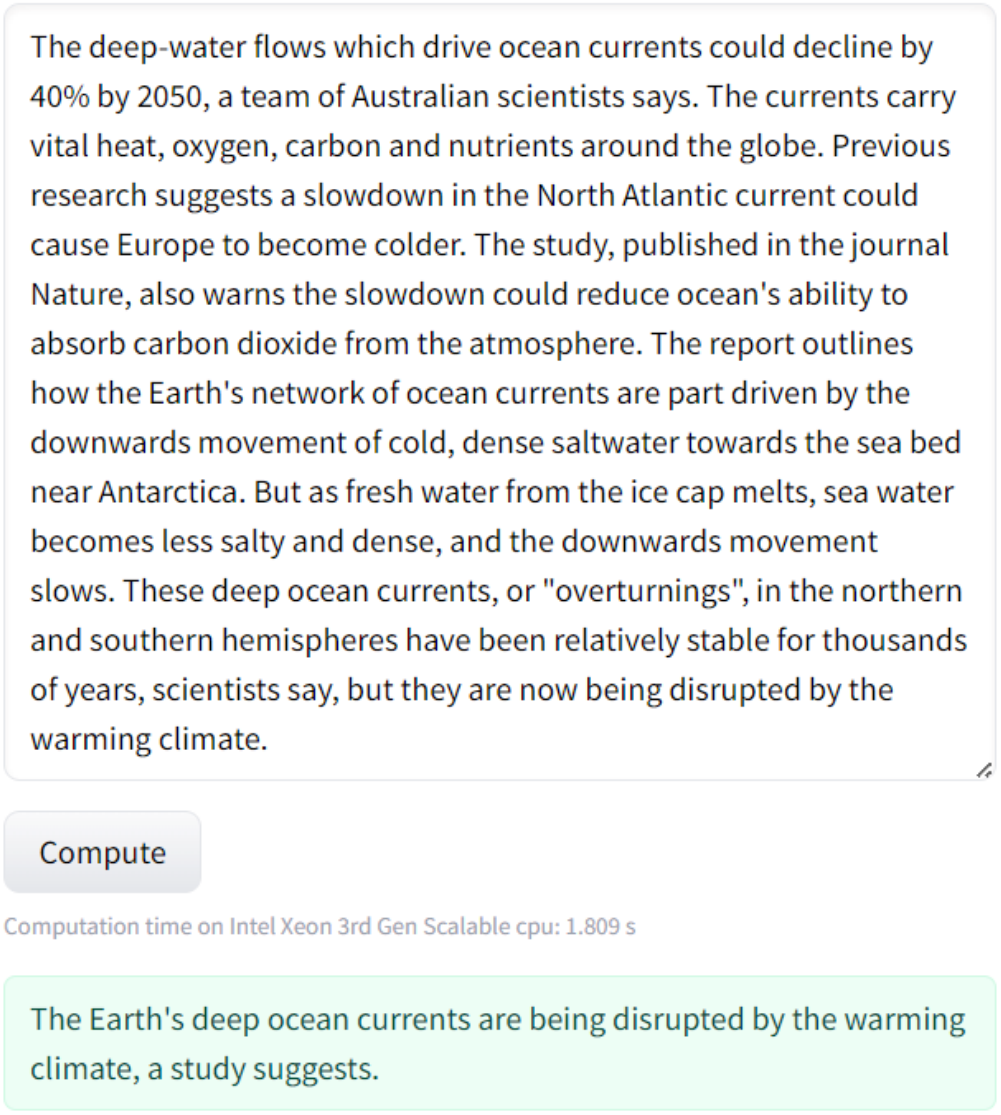
Dataset

- **XSUM (eXtreme Summarization)** is a benchmark dataset for abstractive text summarization developed by researchers at the University of Edinburgh. The dataset consists of approximately **227,000 news articles and their corresponding summaries**, which are written by professional summarizers for a news aggregation service.
- The articles are selected from a diverse range of sources, including **BBC News**, The Independent, and The Guardian, and cover a wide range of topics such as politics, science, and sports. The summaries are typically one or two sentences long and aim to capture the most important information in the original article.
- XSUM is considered to be a challenging dataset for summarization due to the short length of the summaries and the need for the summarizer to generate a concise and coherent summary while preserving the key information in the original article. It has become a widely used benchmark for the evaluation of abstractive summarization models in natural language processing (NLP).

IV. Experiment conduction

References

https://www.bbc.com/news/world-australia-65120327



IV. Experiment conduction

References

https://www.bbc.com/news/world-asia-india-64936519

For years, schools in a drowsy town in Kerala have been facing an unusual problem: students are scarce and teachers have to go out looking for them. They also have to pay from their pockets to bring students to the school. A 150-year-old government upper primary school - which educates students up to the age of 14 - in Kumbanad has 50 students on its rolls, down from about 700 until the late 1980s. Most of them are from poor and underprivileged families who live at the edge of the town. With only seven students, grade seven is the largest class. In 2016, the class had only one student. Getting enough students to the school is a challenge. Each of its eight teachers fork out 2,800 rupees ($34; £28) every month to pay for auto rickshaws (tuk-tuks) ferrying students from home to school and back. They also go door-to-door looking for pupils. Even the few private schools in the area are sending out teachers to look for students - the biggest one has barely 70 students. On a muggy afternoon recently at the upper primary school, you could barely hear the hum of lessons and hubbub of squeals that form the soundscape of a busy schoolhouse. Instead, teachers taught a few children in dark, quiet classrooms. Outside, in the sun-baked courtyard ringing the

Compute

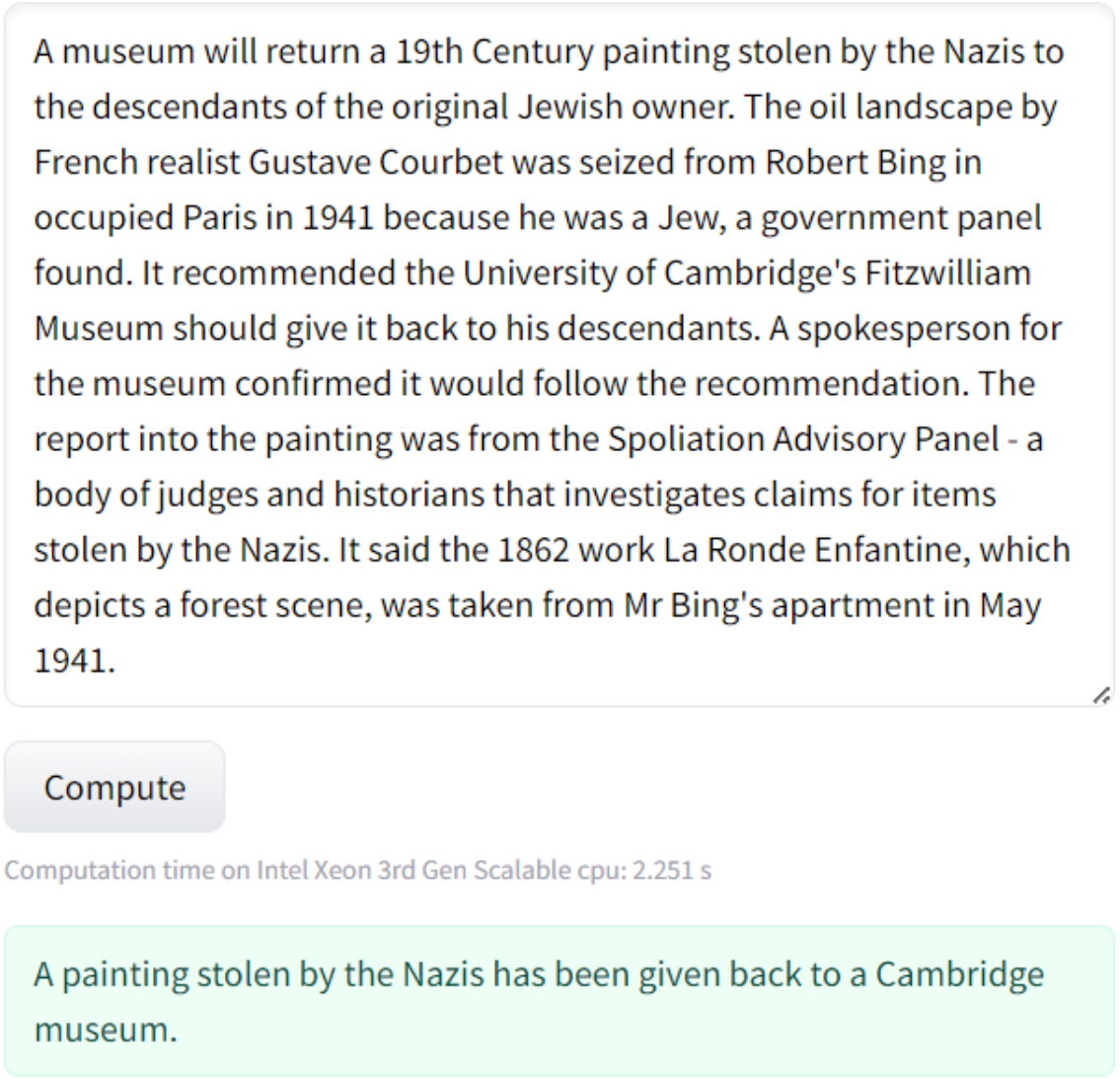
Computation time on Intel Xeon 3rd Gen Scalable cpu: 1.731 s

Schools in India are struggling to get enough students to run their schools.

IV. Experiment conduction

References

https://www.bbc.com/news/uk-england-cambridgeshire-65116330



Training strategy

First round

At first, we tested GPU memory with first 10k samples and batch_size of 16

Data: train/test/validation[10000:1000:1000]

Epoch: 3

Evaluation:


```
{
    'eval_loss': 3.34855318069458,
    'eval_rouge1': 35.1931,
    'eval_rouge2': 13.7162,
    'eval_rougeL': 28.4343,
    'eval_rougeLsum': 28.4329,
    'eval_gen_len': 19.58,
    'eval_runtime': 111.2625,
    'eval_samples_per_second': 8.988,
    'eval_steps_per_second': 2.247,
    'epoch': 3.0
}
```

Training strategy

Second round

In the second round, we doubled everything by picking next 20k samples (no overlapping with first 10k) and the same batch_size of 16, also increase epoch to 5

Data: train/test/validation split[20000:2000:2000]
Epoch: 5

Evaluation:

```
{
    'eval_loss': 3.2764062881469727,
    'eval_rouge1': 36.4663,
    'eval_rouge2': 15.1419,
    'eval_rougeL': 30.0491,
    'eval_rougeLsum': 30.0254,
    'eval_gen_len': 19.619,
    'eval_runtime': 217.6418,
    'eval_samples_per_second': 9.189,
    'eval_steps_per_second': 2.297,
    'epoch': 5.0
}
```

> Our draft training seems converged but has not achieved the SOTA point stated in the paper yet. Stay tuned for round 3

Training strategy

Round 3

Data: train/test/validation split[70000:7000:7000]

Epoch: 5

```
{
    'eval_loss': 3.1328420639038086,
    'eval_rouge1': 37.3896,
    'eval_rouge2': 16.406,
    'eval_rougeL': 30.8594,
    'eval_rougeLsum': 30.8619,
    'eval_gen_len': 19.6073,
    'eval_runtime': 656.091,
    'eval_samples_per_second': 10.669,
    'eval_steps_per_second': 1.334,
    'epoch': 3.0
}
```

Testing

How to use

Here is how to use and start fine-tuning this model on more data:

```
from transformers import AutoModelForSeq2SeqLM,
AutoTokenizer
from transformers import pipeline

checkpoint = 'harouzie/bart-base-xsum'
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
model = AutoModelForSeq2SeqLM.from_pretrained(checkpoint)
```

```
news = """
At least 13 people died after a magnitude 6.8 earthquake
struck southern Ecuador on Saturday afternoon, according to
government officials.

The earthquake struck near the southern town of Baláo and
was more than 65 km (nearly 41 miles) deep, according to the
United States Geological Survey.

An estimated 461 people were injured in the quake, according
to a report from the Ecuadorian president's office. The
government had previously reported that 16 people were
killed but later revised the death toll.

In the province of El Oro, at least 11 people died. At least
one other death was reported in the province of Azuay,
according to the communications department for Ecuador's
president. In an earlier statement, authorities said the
person in Azuay was killed when a wall collapsed onto a car
and that at least three of the victims in El Oro died when a
security camera tower came down.
"""

summarizer = pipeline(task="summarization", model=model,
tokenizer=tokenizer)

summarizer(news)
```

References

1. http://blog.fpt-software.com/text-summarization-in-machine-learning
2. Our model on HuggingFace Hub: https://huggingface.co/harouzie/bart-base-xsum
3. https://sh-tsang.medium.com/brief-review-bart-denoising-sequence-to-sequence-pre-training-for-natural-language-generation-385bf1f43579

BART

Our model on HuggingFace Hub:

- https://huggingface.co/harouzie/bart-base-xsum