Đại học Quốc gia Thành phố Hồ Chí Minh

Trường Đại học Khoa học Tự nhiên

ĐỒ ÁN
TEXT SUMMARIZATION

Nhóm sinh viên thực hiện:

20120364 - Phạm Phước Sang

20120304 - Phan Trần Khanh

Giảng viên hướng dẫn: TS. Ngô Minh Nhựt

Học phần Thống kê học

Lớp 20TN

06 - 2023

Mục Lục

1. Giới thiệu bài toán

Trong bài toán NLP Text Summarization, nhóm chúng em sẽ tìm hiểu cách tóm tắt bản tin tiếng Anh sử dụng mô hình pretrained T5-small của HuggingFace [1], nhằm giúp người đọc hiểu được thông tin chính và cung cấp một cái nhìn tổng quan về nội dung bản tin một cách nhanh chóng.

2. Bộ dữ liệu

2.1. Giới thiệu

XLSum [2] là một tập dữ liệu đa dạng bao gồm tổng cộng 1,35 triệu bài viết từ BBC (đến tháng 08/2021) được annotate article-summary cẩn thận. Tập dữ liệu bao gồm 45 ngôn ngữ có nguồn tài nguyên dữ liệu từ thấp đến cao, trong đó có nhiều ngôn ngữ không có tập dữ liệu công khai sẵn có.

Language	#Samples	Language	#Samples	Language	#Samples
Amharic	5,461	Korean	4,281	Somali	5,636
Arabic	40327	Kyrgyz	2,315	Spanish	44,413
Azerbaijani	7,332	Marathi	11,164	Swahili	10,005
Bengali	8,226	Nepali	5,286	Tamil	17,846
Burmese	5,002	Oromo	5,738	Telugu	11,308
Chinese	39,810	Pashto	15,274	Thai	6,928
English	301,444	Persian	25,783	Tigrinya	4,827
French	9,100	Pidgin[a]	9,715	Turkish	29,510
Gujarati	9,665	Portuguese	23,521	Ukrainian	57,952
Hausa	6,313	Punjabi	8,678	Urdu	40,714
Hindi	51,715	Russian	52,712	Uzbek	4,944
Igbo	4,559	Scottish Gaelic	1,101	Vietnamese	23,468
Indonesian	44,170	Serbian (Cyrillic)	7,317	Welsh	11,596
Japanese	7,585	Serbian (Latin)	7,263	Yoruba	6,316
Kirundi	5,558	Sinhala	3,414	**Total**	**1,005,292**

Hình 1: Ngôn ngữ thuộc bộ dữ liệu XLSum năm 2021[3]

Trong bài toán này, chúng em đã đưa ra quyết định sử dụng subnet English của XL-Sum để xử lý dữ liệu với tổng số lượng hơn 330 nghìn mẫu tin. Tập dữ liệu này bao gồm năm trường thông tin, bao gồm *id, url, title, summary, text*, trong đó trường *summary* là bản tóm tắt của trường *text*. Tuy nhiên, do hạn chế về tài chính, chúng em đã phải đưa ra quyết định giảm 70% subnet English, nhằm giảm thiểu các chi phí phần cứng và tối ưu quá trình huấn luyện dữ liệu, để đạt được hiệu quả tốt nhất trong quá trình nghiên cứu.

Dataset có dạng như sau:

```
{'id': 'uk-wales-56321577',
 'url': 'https://www.bbc.com/news/uk-wales-56321577',
 'title': 'Weather alert issued for gale force winds in Wales',
 'summary': 'Winds could reach gale force in Wales with stormy weather set
   to hit the whole of the country this week.',
 'text': 'The Met Office has issued a yellow weather warning for wind
   covering Wales and England, starting from 21:00 GMT on Wednesday evening.
    Travel and power are both likely to be disrupted, with the warning to
   remain in place until 15:00 on Thursday. Gusts of 55mph (88kmh) are
   likely and could hit up to 70mph on coasts and hills, with heavy and
   blustery showers.'}
```

2.2. Cách phân chia dữ liệu

Ở subnet English của bộ dữ liệu, đã được nhóm đăng tải cố định, tập train có khoảng 307 nghìn mẩu tin, tập test và tập validation đều có khoảng 11.5 nghìn mẩu tin. Sau quá trình thu giảm bộ dữ liệu để đảm bảo phần cứng thì tập train có 214565 mẩu tin, tập test và tập validation đều có 8074 mẩu tin.

Nhóm chúng em chọn 70% dữ liệu để huấn luyện nhằm giảm thiểu các chi phí phần cứng và tối ưu quá trình huấn luyện dữ liệu và sau do mô hình đã hội tụ, việc tăng thêm dữ liệu cũng không giúp cải thiện đáng kể mô hình.

3. Độ đo thích hợp

Để thực hiện việc đánh giá mô hình sau khi đã tinh chỉnh (fine-tuning) trên mô hình T5-small được huấn luyện sẵn (pretrained), nhóm em đã thảo luận và chọn ra các phép độ đo sau:

3.1. ROUGE

Recall-Oriented Understudy for Gíting Evaluation (ROUGE) [4] gồm nhiều độ đo có thể đánh giá chất lượng của văn bản được tạo ra từ mô hình một cách "tự động" nhờ vào số liệu, thông qua việc so sánh văn bản do mô hình tạo với văn bản do con người viết. Phép tính của ROUGE phụ thuộc vào việc đếm những phần tử trùng lặp nhau (như *(n-gram)*[5], chuỗi từ) giữa văn bản được tổng hợp bởi máy và văn bản được tóm tắt bởi con người. Tuy ROUGE sở hữu 4 độ đo là: ROUGE-N, ROUGE-L, ROUGE-W và ROUGE-S, nhưng nhóm em chỉ sử dụng 2 độ đo ROUGE-N và ROUGE-L mà nhóm em cho là phù hợp nhất cho bài toán tóm tắt bản tin.

3.1.1. ROUGE-N: N-gram Co-Occurrence Statistics

Rouge-N, trong đó "N" tương ứng số lượng từ liên kề nhau được sử dụng trong quá trình đánh giá, là một độ đo dựa trên recall sử dụng phép đếm n-gram (từ đơn-từ phức) trên bản tóm tắt do mô hình sinh ra (gọi là candidate summary) và (tập) tóm tắt mẫu (gọi là reference summary) để tính toán.

Rouge-N được phát biểu với công thức như sau:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

Với n là độ dài của từ, cụm từ ($gram_n$) và $Count_{match}(gram_n)$ là số lượng các n-gram đồng thời xuất hiện trong bản tóm tắt được sinh ra và (tập) tóm tắt mẫu.

Mẫu số trong công thức ROUGE-N là tổng số từ trong (tập) tóm tắt mẫu và giá trị này sẽ tăng khi ta có nhiều tóm tắt mẫu cho một bản tóm tắt được sinh ra bởi mô hình. Khi đó, ta sẽ có một tập tóm tắt mẫu (gọi là a set of reference summaries). Bên cạnh đó, tử số cũng sẽ duyệt qua toàn bộ các bản tóm tắt mẫu của ứng viên. Việc cân nhắc một nội dung có thể có nhiều bản tóm tắt mẫu là hoàn toàn hợp lý và đồng thời giúp cho quá trình đánh giá mô hình được khách quan hơn.

Nếu chúng ta sử dụng tập tóm tắt mẫu trong quá trình huấn luyện mô hình thì giá trị ROUGE-N sẽ là giá trị lớn nhất trong tập các giá trị ROUGE-N(r_i, s) với r_i là một bản tóm tắt mẫu và s là bản tóm tắt ứng viên:

$$\text{ROUGE-N}_{multi} = \arg\max_i \text{ROUGE-N}(r_i, s)$$

Bài toán mà nhóm em lựa chọn là tóm tắt bản tin, do đó mô hình nhóm em tạo ra cần phải nắm bắt thông tin cần thiết của bản tin gốc. Vậy nên ROUGE-N sẽ hỗ trợ tốt cho việc đánh giá xem các từ, cụm từ quan trọng có trong bản tóm tắt mẫu có được xuất hiện trong bản tóm tắt đã được tạo ra hay không, vì ROUGE-N sẽ đo lường sự trùng lặp của n-grams, đảm bảo rằng bản tóm tắt được tạo cũng có được nội dung chính tương tự như (các) bản tóm tắt mẫu. Bên cạnh đó, ROUGE-N có ROUGE-1 (unigram), ROUGE-2 (bigram), ... giúp khuyến khích việc đánh giá bản tóm tắt được tạo ra xem nó có mạch lạc, trôi chảy và đúng ngữ pháp không.

3.1.2. ROUGE-L: Longest Common Sequence

ROUGE-L là một độ đo dựa trên F-score được xây dựng trên bài toán dãy con chung dài nhất. Ở bài toán này, ROUGE-L sẽ tính toán trên (tập) bản tóm tắt mẫu và bản tóm tắt do mô hình sinh ra, nếu chúng có chuỗi con chung càng dài thì khả năng chúng có sự tương đồng nhau về nội dung càng cao. Vì vậy đấy cũng là lý do nhóm em chọn độ đo này để đánh giá cho mô hình T5-small.

Sentence level (ROUGE-L) Ở ROUGE-L (sentence level), các bản tóm tắt sẽ được xem là một câu duy nhất, tức bỏ qua các dấu chấm kết thúc câu. Ta xét hai chuỗi từ X và Y có độ dài lần lượt là m và n, với chuỗi X là câu tóm tắt mẫu và Y là câu tóm tắt được tạo ra bởi mô hình, ROUGE-L được phát biểu với công thức như sau:

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

$$\text{ROUGE-L} = F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$$

Trong đó, *LCS(X,Y)* là chuỗi con chung dài nhất (LCS) của X và Y, và β là trọng số thể hiện tầm quan trọng của R_{lcs} và P_{lcs} (tức precision và recall).

Ta nhận thấy ROUGE-L 3.1.2 sẽ mang giá trị 1 khi chuỗi X = chuỗi Y, và ngược lại ROUGE-L sẽ mang giá trị 0 khi *LCS(X,Y)* = 0 (không có chuỗi con chung giữa X và Y).

Có một lợi thế của việc sử dụng LCS trong tính toán, đánh giá mô hình là nó không yêu cầu các từ, cụm từ phải trùng nhau liền tiếp mà chỉ cần theo đúng trình tự trong một câu. Đồng thời mặc định ROUGE-L sẽ lấy chuỗi con chung dài nhất mà không cần thiết lập độ dài của chuỗi trong lúc tính toán.

Bằng việc tính toán trên từ đơn (1-gram - unigram) trùng nhau theo đúng trình tự giữa câu tóm tắt mẫu và câu tóm tắt ứng viên thì ROUGE-L sẽ giúp đánh giá tính tự nhiên của cấu trúc câu văn, xét ví dụ sau:

S1. police killed the gunman
S2. police kill the gunman
S3. the gunman kill police

Với độ đo ROUGE-2 (N=2), ta thấy câu S2 và S3 (các chuỗi ứng viên) đều tồn tại cụm từ "the gunman" có trong S1 (chuỗi mẫu). Theo đánh giá, ta có thể thấy S2 và S3 có thể có điểm ROUGE-2 giống nhau nhưng nghĩa của hai câu hoàn toàn khác nhau. Với độ đo ROUGE-L, cho $\beta = 1$, S2 có giá trị 3/4 *("police the gunman")* và S3 có giá trị 2/4 *("the gunman")* nên S2 sẽ tốt hơn S1 theo đánh giá của ROUGE-L.

Tuy nhiên, trong trường hợp sau:

S4. the gunman police killed

Có thể thấy chuỗi dài nhất là *"the gunman"* hoặc *"police killed"* nên S4 khi đó có giá trị ROUGE-L như S3 (0.5). Tuy nhiên, với ROUGE-2 của S4 sẽ có giá trị là 1.

Summary level (ROUGE-LSUM) Gần tương tự như ROUGE-L, tuy nhiên độ do ROUGE-LSUM sẽ xem xét các bản tóm tắt có từng câu riêng biệt. [6]

Để tính toán ROUGE-LSUM cho bài toán tóm tắt, chúng ta sử dụng hợp của các chuỗi con chung dài nhất (LCS) giữa một câu trong bản tóm tắt mẫu (r_i) với mọi câu trong bản tóm tắt ứng viên (c_j) (tức là toàn bộ bản tóm tắt ứng viên C). Ta xét một bản tóm tắt mẫu có tất cả m từ và u câu và bản tóm tắt ứng viên có tất thảy n từ và v câu.

Rouge-L được phát biểu với công thức như sau:

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n}$$

$$\text{ROUGE-LSUM} = F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

Trong đó, β là trọng số thể hiện tầm quan trọng của R_{lcs} và P_{lcs} (tức precision và recall). $LCS_{\cup}(r_i, C)$ là hợp của các chuỗi con chung dài nhất giữa các câu trong bản tóm tắt mẫu (r_i) với bản tóm tắt ứng viên (C). Giả sử $r_i = w_1 w_2 w_3$, C gồm hai câu $c_1 = w_1 w_4$ và $c_2 = w_1 w_2 w_5$ thì $LCS_{\cup}(r_i, C) = \frac{len(w_1 w_2)}{len(r_i)} = \frac{2}{3}$

3.2. Hàm mất mát (Loss Function)

Mô hình T5-small [7] sử dụng hàm mất mát là cross-entropy loss, vốn được nhiều mô hình ngôn ngữ khác sử dụng để tối ưu quá trình huấn luyện.

Xem xét một chuỗi các từ $x^{(1)}, ..., x^{(T)}$. Ta tính toán phân bố xác suất (probability distributions) $\hat{y}^{(t)}$ ở mỗi bước t, nghĩa là dự đoán phân bố xác suất của mọi từ so với những từ đã cho (label).

Hàm mất mát tại bước t sẽ là kết quả cross-entropy giữa phân bố xác suất $\hat{y}^{(t)}$, và từ chính xác theo mẫu $y^{(t)}$ (trong đó $x^{(t+1)}$ được encode dưới dạng one-hot):

$$J^{(t)}(\theta) = CE(y^{(t)}, \hat{y}^{(t)}) = -\sum_{\omega \in V} y_{\omega}^{(t)} \log \hat{y}_{\omega}^{(t)} = -\log \hat{y}_{x^{(t+1)}}^{(t)}$$

Để tính overall loss cho cả quá trình huấn luyện:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} J^{(t)}(\theta) = \frac{1}{T} \sum_{t=1}^{T} -\log \hat{y}_{x^{(t+1)}}^{(t)}$$

Để hiểu rõ hơn về hàm chi phí, ta có thể giả sử ví dụ sau:

Đưa các câu đầu vào cho mô hình như sau:

```
Input="Hello how are you"
Label="Hi, how do you do"
```

Đầu tiên, ta cần tokenize chúng bằng T5 tokenizer. Giả sử một câu được tokenize thành từng từ tương ứng sẽ là một token duy nhất, đồng thời thêm token đặc biệt của T5 (</s> - để chỉ kết thúc một chuỗi), như vậy chúng ta có được

```
input tokens = [Hello, how, are, you, </s>]
label tokens = [Hi, how, do, you, do, </s>]
```

Sau đó, ta sẽ ánh xạ các token này thành các token ids, bằng cách sử dụng embedding matrix. Ta có được giả định như sau:

```
input_ids = [331, 149, 33, 25, 1]
labels = [483, 149, 55, 25, 55, 1]
```

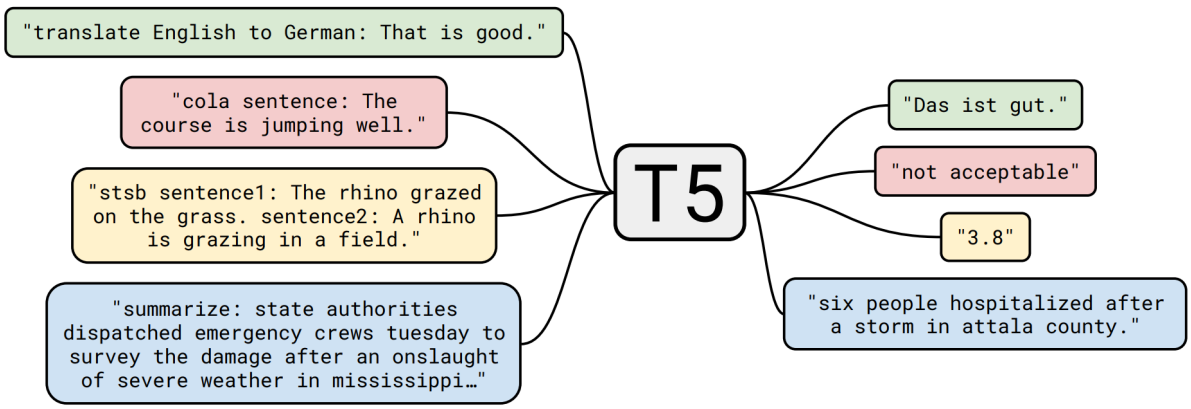
Tiếp theo, ta cho *input_ids* vào T5 encoder và chuyển thành các trường (batch_size, seq_len, hidden_size). Sau đó, T5 decoder sẽ thực hiện như sau:

1. T5 sẽ padding token id đầu tiên cho *input_ids*
2. Với mỗi token trong chuỗi dự đoán, T5 decoder sẽ dự đoán token kế tiếp. Sau đó, T5 decoder sử dụng hàm mất mát cross-entropy để tính toán giữa token dự đoán và token đích tương ứng để đánh giá mô hình.

4. Mô hình T5-small

4.1. Sơ lược

T5-small là một mô hình về ngôn ngữ tự nhiên text-to-text có hỗ trợ checkpoint với khoảng 60 triệu tham số đầu vào. T5-small có hỗ trợ các ngôn ngữ như tiếng Anh, tiếng Pháp, tiếng Rumani và tiếng Đức. T5-small là một trong năm biến thể của T5 với cùng các khả năng thực thi xử lý trên ngôn ngữ tự nhiên như dịch máy, tóm tắt tài liệu, trả lời câu hỏi và các tác vụ phân loại (ví dụ: sentiment analysis). Bên cạnh đó là ta có thể áp dụng T5-small cho các tác vụ hồi quy bằng cách đào tạo nó để dự đoán biểu diễn chuỗi của một số thay vì chính số đó.



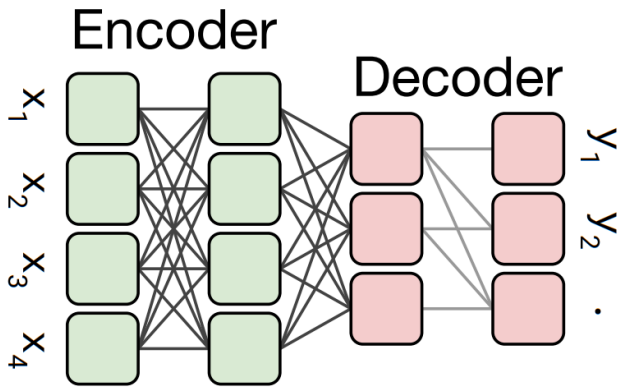
Hình 2: Mô hình T5 [7]

Đầu tiên, một chuỗi token đầu vào được ánh xạ tới một chuỗi embeddings, sau đó được đưa vào encoder. Encoder bao gồm một chồng nhiều "blocks", mỗi block bao gồm hai thành phần phụ: một lớp self-attention và theo sau là một mạng small feed-forward network. Layer normalization phiên bản được tinh giản sẽ được áp dụng cho đầu vào của từng thành phần phụ kể trên. Sau đó sử dụng kỹ thuật residual skip connection để nối đầu vào của từng thành phần phụ trên vào đầu ra của chính nó. Bên cạnh đó, kĩ thuật dropout cũng được vận dụng trong small feed-forward network, skip connection, trọng số attention, và đầu vào và đầu ra của encoder.

Decoder có cấu trúc tương tự như encoder ngoại trừ việc nó sử dụng cơ chế attention tiêu chuẩn sau mỗi lớp self-attention đã tham gia vào đầu ra của encoder. Cơ chế self-attention trong encoder cũng sử dụng một dạng tự hồi quy (autoregressive) hoặc causal self-attention, chỉ cho phép mô hình chú ý đến các đầu ra trong quá khứ. Đầu ra của block cuối cùng của decoder được đưa vào một layer với đầu ra softmax có trọng số được chia sẻ với ma trận embeddings đầu vào. Tất cả các cơ chế attention trong Transformer được chia thành các "heads" độc lập có đầu ra được nối với nhau trước khi được xử lý tiếp.

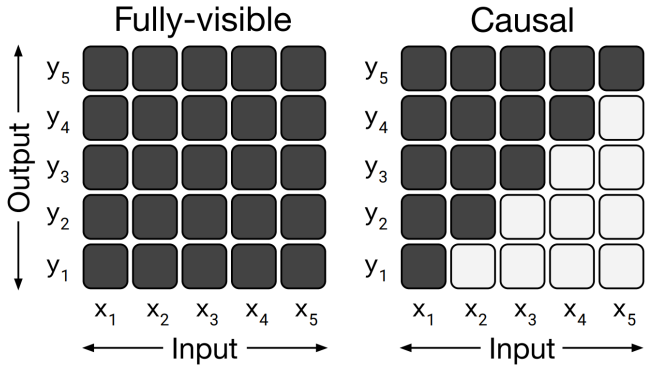
4.2. Kiến trúc Encoder-Decoder

Mặc dù nhóm tác giả của mô hình ngôn ngữ T5 Text-To-Text Transfer Transformer trong quá trình nghiên cứu về transfer learning đã xem xét nhiều biến thể kiến trúc của Transformer, nhưng họ nhận thấy biến thể encoder-decoder gốc hoạt động tốt nhất với text-to-text framework của họ. Dù kiến trúc encoder-decoder này sử dụng gấp đôi số tham số so với kiến trúc "encoder-only" (ví dụ: BERT) hoặc "decoder-only" (ví dụ: Mô hình ngôn ngữ - LM), nó lại có chi phí tính toán tương tự. Ngoài ra việc chia sẻ các tham số trong encoder và decoder cũng được nhóm tác giả chứng minh là không làm giảm hiệu suất đáng kể dù giảm một nửa tổng số lượng tham số.



Hình 3: Kiến trúc encoder-decoder sử dụng fully-visible masking cho encoder và encoder-decoder attention, với causal masking cho decoder

Với fully-visible masking, nó cho phép cơ chế self-attention hoạt động trên toàn bộ câu đầu vào ở mọi khoảng thời gian sinh các phần tử đầu ra. Với causal masking, nó ngăn không cho phần tử đầu ra thứ i phụ thuộc vào bất kỳ phần tử đầu vào nào từ "tương lai" (tức lớn hơn i).



Hình 4: Ma trận đại diện cho các mẫu masking attention khác nhau. Đầu vào và đầu ra của cơ chế self-attention lần lượt được ký hiệu là x và y. Mỗi ô đen ở hàng i và cột j cho biết rằng cơ chế self-attention được phép tham gia vào phần tử đầu vào j ở thời điểm sinh đầu ra i. Mỗi ô trắng chỉ ra rằng cơ chế self-attention không được phép tham gia vào cặp i, j tương ứng

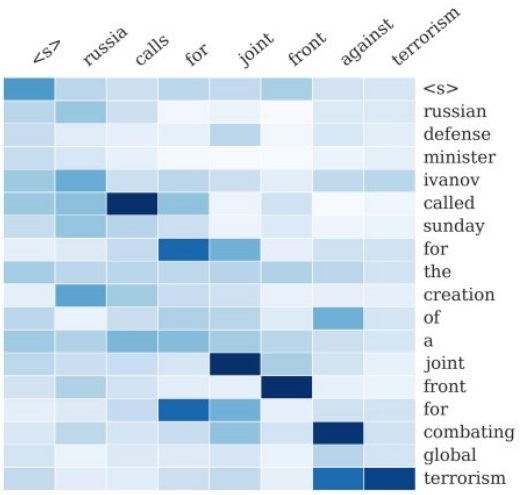
Kiến trúc encoder-decoder bao gồm hai lớp: encoder (được cung cấp một chuỗi đầu vào) và decoder (tạo ra một chuỗi đầu ra mới), như được thể hiện ở Hình 3. Lớp encoder sẽ sử dụng fully-visible masking vì một số ngữ cảnh được cung cấp cho mô hình trước đó mà sau này cần được sử dụng để đưa ra dự đoán và đồng thời giúp đánh giá mức độ tương quan giữa các token không lân cận nhau tốt hơn. Còn lớp decoder sử dụng causal masking trong quá trình huấn luyện để mô hình không thể "nhìn thấy tương lai" vì nó đang tạo ra kết quả đầu ra và làm mô hình phụ thuộc vào đầu ra mà nó đã tạo để sinh thêm nhằm tăng tính mạch lạc của câu đầu ra.

4.3. Attention

Cơ chế attention có thể được hiểu là một phép ánh xạ từ một query và một tập hợp các cặp key-value sang một giá trị đầu ra, trong đó query, key, value và giá trị đầu ra đó đều là các vector. Đầu ra là kết quả dưới dạng tổng trọng số của values, trong đó trọng số được gán cho từng value dựa trên hàm tương thích (compatibility function) của cặp query và key. [8]

Ngoài ra, cơ chế attention cũng được hiểu nôm na là một ma trận tương đồng (similarity matrix) của một chuỗi. Vì chúng ta cần quan tâm đến các phần tử quan trọng trong chuỗi thông qua việc xét từng phần tử, vậy nên chúng ta cần attention-vector cho từng phần tử. Nếu các phần tử trong chuỗi có nhiều hơn một attention-vector thì chúng ta nhân các vector này với nhau rồi sử dụng giá trị trọng số trung bình cho attention-vector cuối, khi này đây được coi là multi-headed attention. [8]

Với cơ chế attention, mô hình sẽ không phải học toàn bộ chuỗi đầu vào (điều sẽ dẫn đến bất lợi nếu đầu vào là một chuỗi quá dài) mà tập trung vào các token thích hợp với ngữ cảnh, đồng thời trong quá trình sinh token đầu ra mô hình cũng có khả năng tận dụng những token liên quan nhất có được từ đầu vào.



Hình 5: Ví dụ về kết quả khi tính toán trên vector attention [9]

Với vector trừu tượng Query, Key và Value thông qua việc biến đổi tuyến tính của đầu vào, tức là nhân vector phần tử với ba ma trận W_q, W_k và W_v. Với mỗi phần tử trong chuỗi, chúng ta có một vector Q, K và V tương ứng, được sử dụng để tính toán các vector cho từng phần tử.

Trong đó:

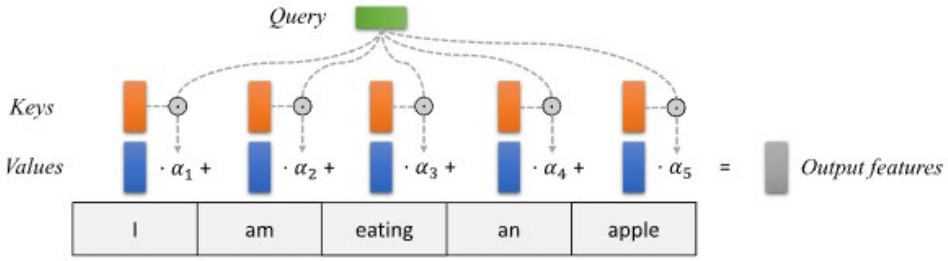
- Q: Vector (đầu ra của lớp tuyến tính) liên quan đến những gì mã hóa (đầu ra, có thể là đầu ra của lớp mã hóa hoặc lớp giải mã). Q là vector thể hiện cái chúng ta đặt trọng tâm trong chuỗi.
- K: Vector (đầu ra của lớp tuyến tính) liên quan đến những gì sử dụng làm đầu vào cho đầu ra. K là vector giúp chúng ta xác định được phần tử nào cần để ý.
- V: Vector được học (đầu ra của lớp tuyến tính) là kết quả của các phép tính, liên quan đến đầu vào.

Khi đó, công thúc chung như sau:

$$c = \sum_{i=1}^{n} sim_p \left(W_k k^i, W_q q\right) W_v v^i,$$
$$W_q \in \mathbb{R}^{n_m \times m_q}, W_k \in \mathbb{R}^{n_m \times m_k}, W_v \in \mathbb{R}^{n_v \times m_v}$$

Tiếp theo, để đánh giá những yếu tố cần trọng tâm, hàm tính điểm (score) *sim*. Với hàm điểm, sẽ lấy một Query và Key làm đầu vào và tạo ra điểm/ trọng số chi cặp query-key.Thường các hàm này sẽ được cài đặt bởi các độ đo tương tự đơn giản như tích vô hướng, hoặc một mạng nơ-ron nhỏ (MLP), sau đó được truyền qua hàm softmax để tạo ra một phân bố xác suất. Tại đây, hàm tính điểm này được tham số hóa bởi **feed-forward network** 4.4 với một lớp ẩn và mạng này được huấn luyện chung với các phần khác của mô hình.

$$\alpha_i = \frac{\exp f_{attn}\left(key_i, query\right))}{\sum_j \exp f_{attn}\left(key_j, query\right))},$$
$$out = \sum_i \alpha_i \cdot value_i$$



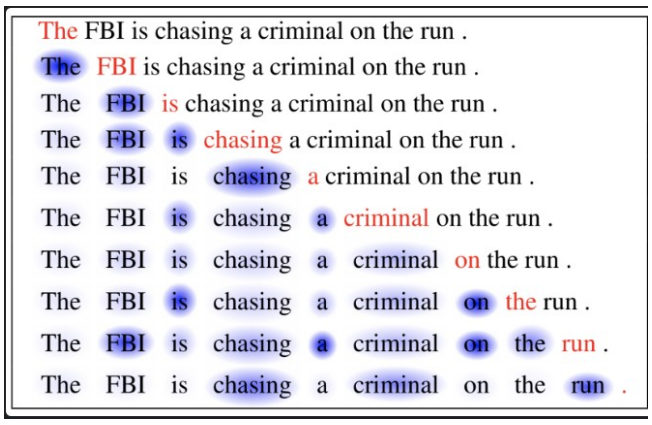
Hình 6: Ví dụ về hàm attention trong việc tính toán các trọng số trong chuỗi [8]

4.3.1. Self-attention

Block nền tảng của mô hình Transformer là cơ chế self-attention. Self-attention, còn được gọi là intra-attention, là một cơ chế attention chỉ được sử dụng cho một câu. Cơ chế này cho phép tạo ra một ma trận với hàng và cột đều là cùng một câu, giúp hiểu được sự liên quan giữa các phần tử khác nhau trong câu.

Self-attention đã được chứng minh là rất hiệu quả trong các ứng dụng như tóm tắt văn bản, tạo chú thích cho hình ảnh, đọc máy và các ứng dụng khác. Self-attention cùng với kiến trúc transformer đã thay thế hoàn toàn kiến trúc mạng nơ-ron hồi tiếp RNN bằng các mô hình fully connected và vẫn mang lại kết quả rất tốt. Điều này đánh dấu một cột mốc quan trọng trong việc áp dụng cơ chế attention cho các bài toán về xử lý ngôn ngữ tự nhiên (NLP).

Ví dụ, trong hình dưới đây, các từ đang được xét (chữ đỏ) và các từ được bôi xanh thể hiện sự ảnh hưởng của chúng lên từ màu đỏ [10],

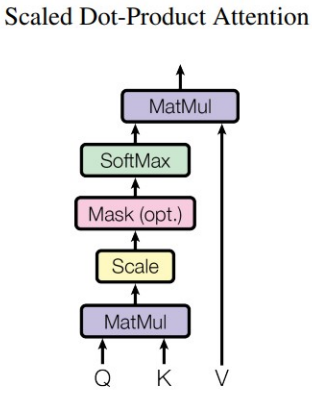


Hình 7: Từ đang xét có màu đỏ và kích thước và độ đậm của sắc xanh biểu thị trọng số attention [10]

4.3.2. Scaled Dot-product Attention

Với tập đầu vào gồm query $Q \in \mathbb{R}^{T \times d_k}$, keys $K \in \mathbb{R}^{T \times d_k}$ và values $V \in \mathbb{R}^{T \times d_k}$ và T là độ dài của chuỗi, d_k và d_v là kích thước ẩn của vector cho các vector query/key và value tương ứng [10]. Tensor attention được biểu diễn như sau:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Hình 8: Cơ chế scaled dot-product attention. [10]

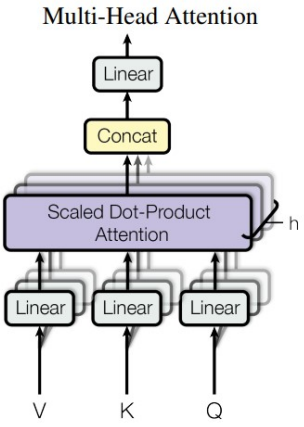
Lưu ý rằng hàm softmax có thể mất đạo hàm với các giá trị đầu vào lớn. Vì vậy, việc chia cho căn bậc hai của kích thước của các vector (tương đương với độ dài Euclid của vector đơn vị của d_k) sẽ ngăn hàm softmax trở nên quá lớn.

4.3.3. Multi-head Attention

Thay vì chỉ tính attention một lần, cơ chế multi-head tính attention được scale bằng tích vô hướng nhiều lần song song. Các đầu ra attention độc lập được ghép nối và được biến đổi tuyến tính để đạt kích thước mong muốn.

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, .., \text{head}_h\right)\mathbf{W}^O \\ \text{head}_i &= \text{Attention}\left(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V\right)\end{aligned}$$

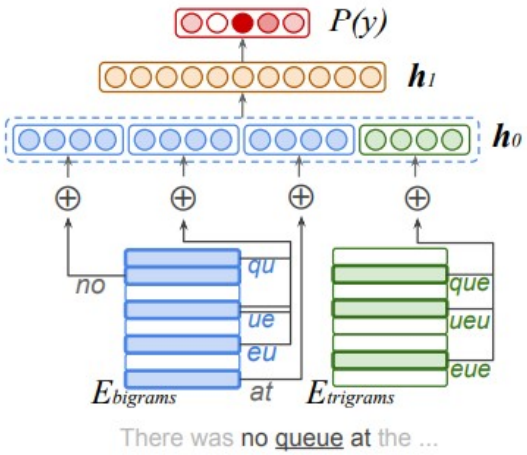
Với các ma trận tham số học là $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ và $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. Ở đây, h là số lượng các lớp attention song song nhau (head). Do kích thước giảm của mỗi head, tổng chi phí tính toán của multi-head attention tương tự như của single-head attention với chiều đầy đủ.



Hình 9: Cơ chế multi-head attention bao gồm nhiều lớp attention chạy song song. [10]

4.4. Small Feed-forward Network

Các mô hình Long Short-Term Memory (LSTM) đã đạt được kết quả tốt với lượng bộ nhớ nhỏ bằng cách sử dụng biểu diễn đầu vào dựa trên ký tự. Tuy nhiên, độ trễ của mô hình tương đối lớn do phải thực hiện quá nhiều phép tính toán ma trận [11]. Feed-forward neural network có tiềm năng chạy nhanh hơn so với LSTM.



Hình 10: Cấu trúc mạng cho mô hình sử dụng bigrams của từ trước, từ hiện tại và từ tiếp theo, và trigrams của từ hiện tại [11]

Các kiến trúc mạng được thiết kế để giới hạn bộ nhớ và thời gian chạy của mô hình 11 mô tả kiến trúc của mô hình:

- Các đặc tính rời rạc được phân theo các nhóm khác nhau ($E_{bigrams}$), với một ma trận $E_g \in \mathbb{R}^{V_g \times D_g}$ với mỗi nhóm.
- Các đặc tính được trích xuất ra tại mọi nhóm được định dạng lại thành một vector và nối với đầu ra của lớp $h_0 = [X_g E_g | \forall g]$.

- Một lớp ẩn, h_1, với M là đơn vị tuyến tính chỉnh (ReLU) [12] kết nối đầy đủ tới h_0.
- Một hàm softmax mô hình hóa xác suất của một lớp đầu ra của lớp y: $P(y)\ \alpha\ \exp\left(\beta_y^T h_1 + b_y\right)$, với $\beta_y \in \mathbb{R}^M$ và b_y là vetor trọng số và bias.

Nhóm đặc trưng riêng biệt g với mỗi n-gram có độ dài là N và áp *Random feature mixing* [13] trực tiếp với kích thước V_g. Giá trị của đặc tính v cho chuỗi n-gram x là $v = H(x) \mod V_g$, với H là hàm băm. Giá trj V_g thuộc khoảng 100-5000, với giá trị nhỏ hơn số mũ của n-gram thô duy nhất.

Random Feature Mixing Việc xây dựng các mô hình tuyến tính trên các đặc trưng được tạo ra từ đầu vào. Các đặc trưng được biểu diễn dưới dạng chuỗi (ví dụ: "w=apple" được hiểu là "chứa từ apple") và được chuyển đổi thành các chỉ số đặc trưng được duy trì bởi bảng chữ cái. Các trường hợp được biểu diễn dưới dạng vector thưa và mô hình là một vector trọng số dày đặc. Tuy nhiên, bảng chữ cái lưu trữ một chuỗi cho mỗi đặc trưng, có thể là mỗi unigram hoặc bigram mà nó gặp phải, nên nó lớn hơn nhiều so với vector trọng số.

Ý tưởng của bài báo [13] là thay thế bảng chữ cái bằng một hàm ngẫu nhiên từ chuỗi sang số nguyên trong khoảng từ 0 đến một kích thước dự định, kích thước này kiểm soát số lượng tham số trong mô hình. Việc va chạm giữa các đặc trưng được kiểm soát bởi kích thước dự định. Chúng ta có thể giảm thiểu đáng kể số lượng va chạm mà không làm hại quá trình học. Ngay cả khi sử dụng không gian đặc trưng vô cùng lớn để tránh va chạm, việc lưu trữ bảng chữ cái vẫn bị loại bỏ.

4.5. Layer Normalization

Phương pháp *layer normalization* là một trong những kỹ thuật được áp dụng để tối ưu hóa tốc độ huấn luyện của các mô hình mạng neural đa dạng. Phương pháp này tính toán trực tiếp các số liệu thống kê chuẩn hóa từ các đầu vào tổng hợp đến các nơ-ron bên trong một lớp ẩn. *Layer normalization* thực hiện chuẩn hóa đầu vào trên các lớp thay vì chuẩn hóa các đặc trưng đầu vào trên từng *batch* như trong *batch normalization* [14]. Phương pháp này được cho là giúp cải thiện tốc độ huấn luyện và chất lượng mô hình của các mạng nơ ron.

Feed-forward neural network là một ánh xạ phi tuyến tính với đầu vào là vector x và đầu ra là vector y. Cho lớp ẩn thứ i trong một mạng thần kinh chuyển tiếp sâu, mạng nơ-ron, a^l là vector đại diện của các đầu vào tổng hợp cho các nơ-ron trong lớp đó. Các đầu vào tổng hợp được tính toán thông qua ánh xạ tuyến tính với ma trận trọng số W^l và đầu vào bottom-up h^l như sau:

$$a_i^l = w_i^{l^\mathsf{T}} h^l$$
$$h_i^{l+1} = f(a_i^l + b_i^l)$$

Trong đó $f(\cdot)$ là một hàm phi tuyến tính cho từng phần tử và w_i^l là ma trận trọng số kế tiếp tại lớp ẩn thứ i, b_i^l là bias vô hướng. Các tham số được tối ưu thông qua thuật toán dựa trên độ dốc, trong đó độ dốc được tính toán bằng phương pháp lan truyền ngược.

Sự thay đổi kết quả đầu ra của một lớp có xu hướng gây ra sự thay đổi tương quan trong tổng các đầu vào cho lớp tiếp theo, đặc biệt là với các đơn vị ReLU mà đầu ra có thể thay đổi nhiều. Điều này cho thấy vấn đề *đổi mới đồng tham số* có thể được giảm bớt bằng cách cố định trung bình và phương sai của tổng các đầu vào trong mỗi lớp. Vì vậy, tính toán các số liệu chuẩn hóa lớp trên tất cả các đơn vị ẩn trong cùng một lớp như sau:

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l$$

$$\sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^l - \mu^l\right)^2}$$

Với H biểu thị số lượng units ẩn trong một lớp.

Không giống như *batch normalization*, *layer normalization* không áp đặt bất kỳ ràng buộc nào về kích thước của một mini-batch và nó có thể được sử dụng với kích thước batch là 1.

Layer normalized recurrent neural networks Các mô hình seq2seq gần đây [15] sử dụng mạng nơ-ron tuần hoàn nhỏ gọn để giải quyết các vấn đề dự đoán liên tục trong xử lý ngôn ngữ tự nhiên (NLP). Thông thường, các nhiệm vụ NLP có độ dài câu khác nhau cho các trường hợp huấn luyện khác nhau. Điều này dễ xử lý trong một mạng RNN vì cùng một trọng số được sử dụng lại cho mỗi bước thời gian xử lý (timestep). Tuy nhiên, khi chúng ta áp dụng *batch normalization* cho một mạng RNN, chúng ta cần tính toán và lưu trữ các số liệu thống kê riêng biệt cho từng bước thời gian trong một chuỗi[16]. Điều này gây khó khăn nếu một chuỗi kiểm tra dài hơn bất kỳ chuỗi huấn luyện nào. *Layer normalization* có thể giải quyết vấn đề này vì các thuật ngữ được chuẩn hóa của nó chỉ phụ thuộc vào tổng các đầu vào cho một lớp tại bước thời gian hiện tại với một tập tham số và bias được sử dụng trên tất cả các bước thời gian.

Với RNN tiêu chuẩn, các đầu vào được tổng hợp trong lớp lặp lại (recurrent) được tính từ đầu vào x^t và vector trạng thái ẩn trước h^{t-1} được tính toán như sau:

$$a^t = W_{hh}h^{t-1} + W_{xh}x^t \tag{1}$$

Layer normalization có thể sử dụng công thức sau để điều chỉnh các giá trị activation tại lớp lặp lại:

$$h^t = f\left[\frac{g}{\sigma^t}\odot\left(a^t - \mu^t\right) + b\right]$$

$$\mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t$$

$$\sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^t - \mu^t\right)^2}$$

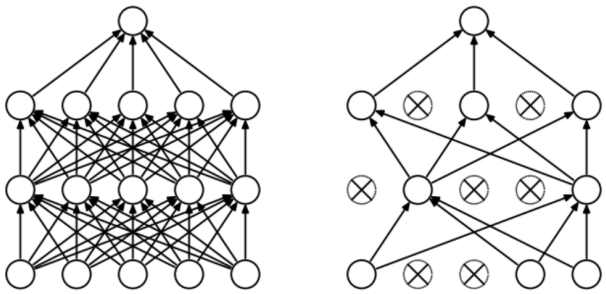
Trong đó, W_{hh} là trọng số ẩn lặp lại (recurrent) và W_xh là giá trị đầu vào bottom-up. b và g lần lượt là bias và tham số khuyết đại có cùng chiều.

4.6. Learning Rate Schedule

Trong quá trình pre-training, nhóm tác giả của T5 đã sử dụng một biểu thức learning rate schedule: $\frac{1}{\sqrt{\max(n,k)}}$, trong đó n là lần lặp để huấn luyện ở hiện tại và k là số bước warm-up (có giá trị là 10^4 trong tất cả các bước thí nghiệm của nhóm tác giả). Khi này learning rate sẽ không đổi, ở giá trị 0,01 cho 10^4 bước đầu tiên, sau đó learning rate giảm dần theo cấp số nhân cho đến khi quá trình huấn luyện pre-training kết thúc.

4.7. Dropout

Dropout với hệ số p nghĩa là trong quá trình train model, với mỗi lần thực hiện cập nhật hệ số trong gradient descent ta ngẫu nhiên loại bỏ p% số lượng node trong layer đấy, hay nói cách khác là dữ lại (1-p%) node. Mỗi layer có thể có các hệ số dropout p khác nhau. Ở mỗi bước khi train model thì ngẫu nhiên (1-p%) các node bị loại bỏ nên model không thể phụ thuộc vào bất kì node nào của layer trước mà thay vào đó có xu hướng trải đều weight, điều này giúp tránh được overfitting.



Hình 11: Hình minh họa dropout

5. Mô tả thí nghiệm

5.1. Thông số cài đặt

Để phục vụ cho thao tác huấn luyện dữ liệu, nhóm em sử dụng các thông số ở Bảng 1

Đối tượng	Tham số	Ý nghĩa	Giá trị
T5 tokenizer input tokenizer	truncation	Cắt dữ liệu tại điểm tối đa	True
	max length	Độ dài tối đa của token	1024
T5 tokenizer label tokenizer	truncation	Cắt dữ liệu tại điểm tối đa	True
	max length	Độ dài tối đa của token	128
Model pretrain	max length	Độ dài tối đa mô hình sinh ra	128
	length penalty	$>$ 0,0 khuyến khích các chuỗi dài hơn, $<$ 0,0 khuyến khích các chuỗi ngắn hơn.	0.6
	no repeat ngram size	$>$ 0, tất cả các ngram có kích thước đó chỉ có thể xảy ra một lần.	2
	num beams	Số lượng beam để dùng beam search	15
Training args	epochs		10
	learning rate		5.10^{-4}
	lr schedule type	Cách thức tăng learning rate	linear
	wramup steps	Tỷ lệ của tổng số bước đào tạo được sử dụng để khởi động tuyến tính từ 0 đến learning_rate.	90
	optimize	Thuật toán cải thiện	adafactor
	weight decay	Giảm dần trọng số	0.01
	per device train batch size	Batch size cho mỗi GPU/TPU core/CPU vào huấn luyện.	2
	per device eval batch size	Batch size cho mỗi GPU/TPU core/CPU vào kiểm tra.	1
	evaluation strategy	Độ đo để kiểm chứng	steps
	eval steps	Số bước để kiểm chứng	100
	predict with generate	Dự đoán với chuỗi được sinh ra	True
	generation max length	Độ dài tối đa chuỗi sinh ra	128
	save steps	Bước để lưu vào check point	1000
	batch size		128
	logging steps	Bước để lưu vào log	10
Predict with test data	num beams	Số lượng beam để dùng beam search	15
	num return sequences	Số lượng câu được trả về	1
	no repeat ngram size	$>$ 0, tất cả các ngram có kích thước đó chỉ có thể xảy ra một lần	1
	remove invalid values	Loại bỏ những giá trị không hợp lệ	True
	max length	Độ dài tối đa câu được sinh ra	128

Bảng 1: Bảng thông số trong quá trình huấn luyện

5.2. Chi tiết thí nghiệm

Trong phần thí nghiệm này, chúng em đã quyết định sử dụng mô hình T5 để thực hiện fine-tune trên dữ liệu mới, dựa trên script [17] và sửa đổi bổ sung. Quá trình thực hiện được thực hiện theo các bước sau:

Đầu tiên, chúng em thu thập dữ liệu từ tập *csebuetnlp/xlsum* 2.2, và chia thành ba tập chính là *train, test* và *validation*. Từ đó, chúng tôi đã có thể tiến hành xây dựng mô hình và

tiếp tục các bước tiền xử lý để chuẩn bị cho quá trình fine-tune.

Sau đó, tiến hành đánh dấu các token cho các câu trong trường *summary* và *text* của dữ liệu, thông qua pretrained của T5-small tokenizer. Các trường dữ liệu khác đã bị loại bỏ để tối ưu quá trình huấn luyện. Khi đó, dữ liệu chỉ còn lại bao gồm ba trường chính là *input ids, attention mask, labels*, giúp cho việc huấn luyện được diễn ra một cách dễ dàng và hiệu quả hơn.

Sau khi tiến hành tiền xử lý dữ liệu như đã đề cập ở trên, chúng em đã thực hiện đo đạc kết quả bằng độ đo Rouge [6], giữa giá trị *label* và *summary*. Để đảm bảo tính chính xác và đáng tin cậy của kết quả, chúng tôi đã tiến hành rút gọn các câu trong dữ liệu bằng cách loại bỏ các kí tự đặc biệt ở cuối câu như !?.;,'" và thêm dấu chấm vào cuối câu.

```
The Met Office has issued a yellow weather warning for wind covering Wales
    and England, starting from 21:00 GMT on Wednesday evening. Travel and
    power are both likely to be disrupted, with the warning to remain in
    place until 15:00 on Thursday. Gusts of 55mph (88kmh) are likely and
    could hit up to 70mph on coasts and hills, with heavy and blustery
    showers.
```

sau khi thực hiện xử lý, cấu trên sẽ được tách thành 5 câu lẻ:

```
The Met Office has issued a yellow weather warning for wind covering Wales
    and England.
starting from 21:00 GMT on Wednesday evening.
Travel and power are both likely to be disrupted.
with the warning to remain in place until 15:00 on Thursday.
Gusts of 55mph (88kmh) are likely and could hit up to 70mph on coasts and
    hills.
with heavy and blustery showers.
```

5.3. Kết quả

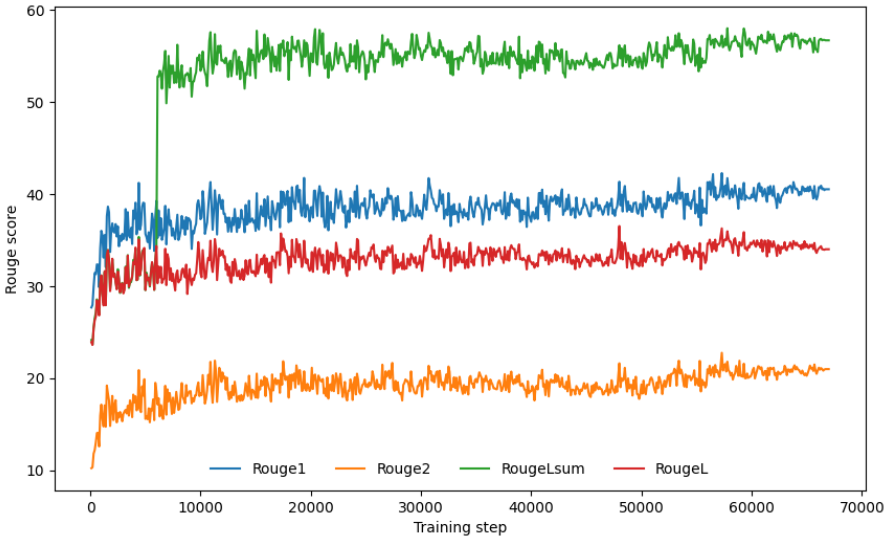
Sau khi huấn luyện dữ liệu bằng dữ liệu mới, nhóm em có kết quả như sau:

Rouge-1	40.47
Rouge-2	20.60
Rouge-L	34.60
Rouge-Lsum	56.60

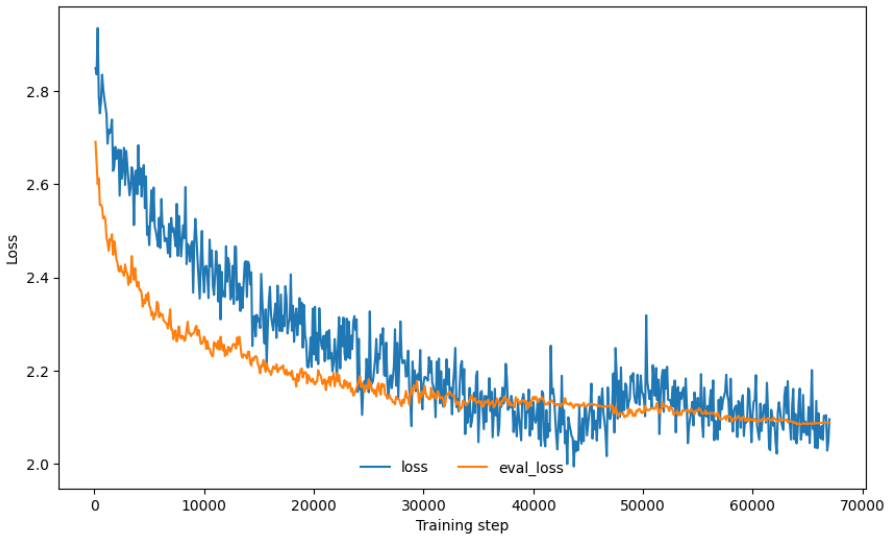
Theo như đồ thị được vẽ 12, chúng em thấy cả 4 độ đo có kết quả tăng khi so sánh với kết quả ban đầu trên tập dữ liệu. Với sự gia tăng rõ rệt tại độ đo Rouge-2 và Rouge-L, ta có thể thấy mô hình sau khi huấn luyện lại trên tập dữ liệu mới có độ trùng lắp về từ ghép (từ đôi) và các đoạn văn có độ dài chung dài nhất giữa câu được sinh ra và câu nhãn (label).

Dựa vào quan sát biên độ tại các độ đo, chúng em nhận thấy rằng giá trị biên độ giảm dần theo từng bước tăng dần. Tuy nhiên, tại khoảng giá trị từ 40000 đến 50000, chúng em quan sát được một giá trị biên độ nhỏ, cho thấy mô hình đã được huấn luyện đến mức độ rouge hội tụ .

Dựa vào quan sát về đồ thị mất mát 13, chúng em nhận thấy rằng chi phí của mô hình giảm dần theo số lượng bước huấn luyện và dần hội tụ về giá trị ổn định ở khoảng step 60000. Điều này cho thấy mô hình đã hội tụ và không cần tiếp tục huấn luyện nữa.



Hình 12: Độ đo của dữ liệu dựa trên bước huấn luyên



Hình 13: Đồ thị chi phí huấn luyện và đánh giá

5.4. Đánh giá mô hình

Bài viết	Dữ liệu nhãn	Kết quả từ mô hình	Đánh giá
Portsmouth sea wall hole repair work delayed	Work to repair a sea wall that collapsed during storms in Portsmouth, has been delayed.	Donald Trump's inauguration speech was not an inaugural address.	Mô hình T5-small tóm tắt sai nội dung chính.
What's happening in Bolivia?	Bolivia's President Evo Morales has now left the country after a controversial election there.	Bolivian President Evo Morales has resigned after almost 14 years in power.	Mô hình T5-small chỉ tóm tắt được nội dung tương đối vì chưa đề cập về "cuộc bầu cử đầy tranh cãi".
India floods: Over 1,000 train passengers rescued near Mumbai	Indian authorities have rescued 1,050 people from a train after it became trapped by flooding near Mumbai.	Passengers stranded in the southern Indian state of Mahalaxmi have been told to stay onboard.	Mô hình T5-small tóm tắt sai nội dung chính về công cuộc cứu hộ, nhưng đề cập về sự kiện chi tiết.
Intu Milton Keynes shopping centre has new operator	A major shopping centre has a new operator after the previous company went into administration.	One of the UK's biggest retail centres is to be rebranded.	Mô hình T5-small chỉ tóm tắt được nội dung tương đối chính xác.
Autism: Parents calls for more support in Northern Ireland schools	Paul McDonald's autistic son, Jim, has been suspended from his mainstream primary school for 30 days in the past three months.	The number of children with autism in Northern Ireland is rising.	Mô hình T5-small tóm tắt nội dung tổng thể thay vì một cá nhân như nhãn đề cập.

Bảng 2: Bảng so sánh kết quả test và label của các bài báo BBC News

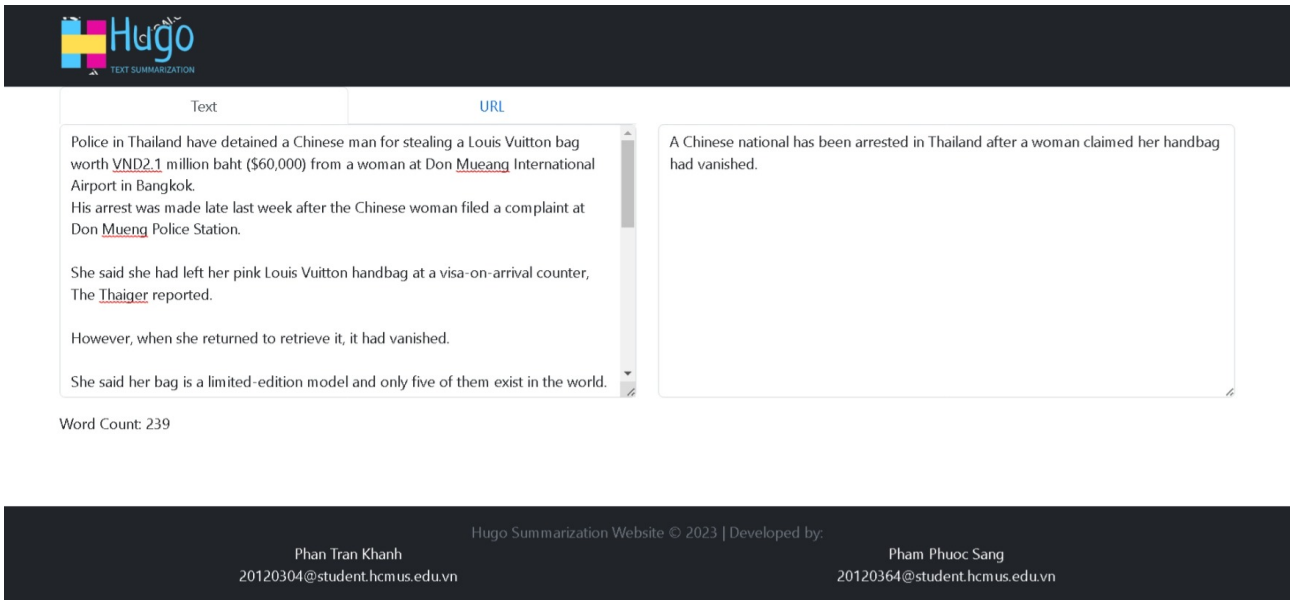
Kết quả từ bảng so sánh cho thấy bản tóm tắt từ mô hình sẽ chú trọng vào kết quả cuối, dẫn tới một số bản tin tuy không đề cập nội dung chính nhưng mô hình sẽ đưa ra kết quả sau sự kiện chính trong bản tin. Vậy nên trong đa số trường hợp mô hình sẽ tóm tắt gần đúng sự kiện chính, tuy nhiên vẫn có trường hợp mô hình sinh ra nội dung văn bản hoàn toàn không liên quan.

Model	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
T5-small(fine-tuned)	**40.47**	**20.60**	**34.60**	**56.60**
mt5-small-finetuned-mt5-en	23.8952	5.8792	18.6495	18.7057
t5-small-finetuned-xlsum-en	23.7508	5.5427	18.6777	18.652
cos801-802-hf-workshop-mt5-small	20.928	6.3239	17.4455	17.4566
t5-small-finetuned-xlsum	15.4289	3.146	12.7682	12.912

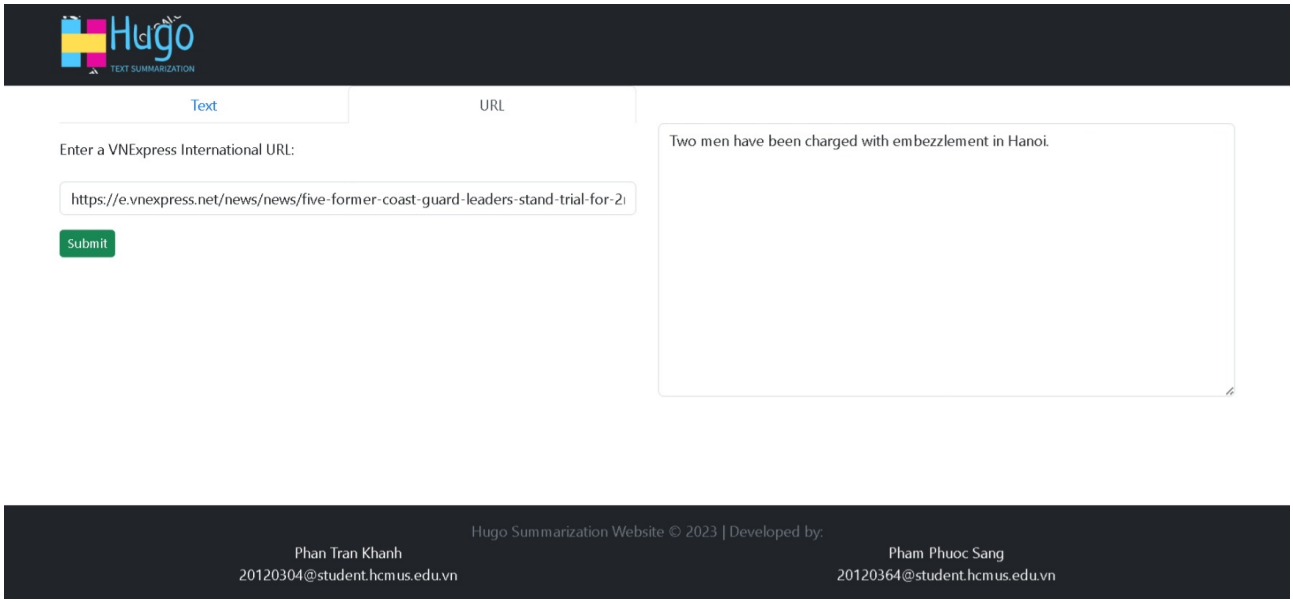
Bảng 3: Bảng kết quả so sánh các mô hình huấn luyện trên tập xlsum

6. Hình ảnh minh họa

Nhóm em đã thực hiện cài đặt ứng dụng web news-summarization-website để biểu diễn mô hình đã fine-tuning. Thầy có thể thực hiện chạy local thông qua một số cài đặt trong README.md.

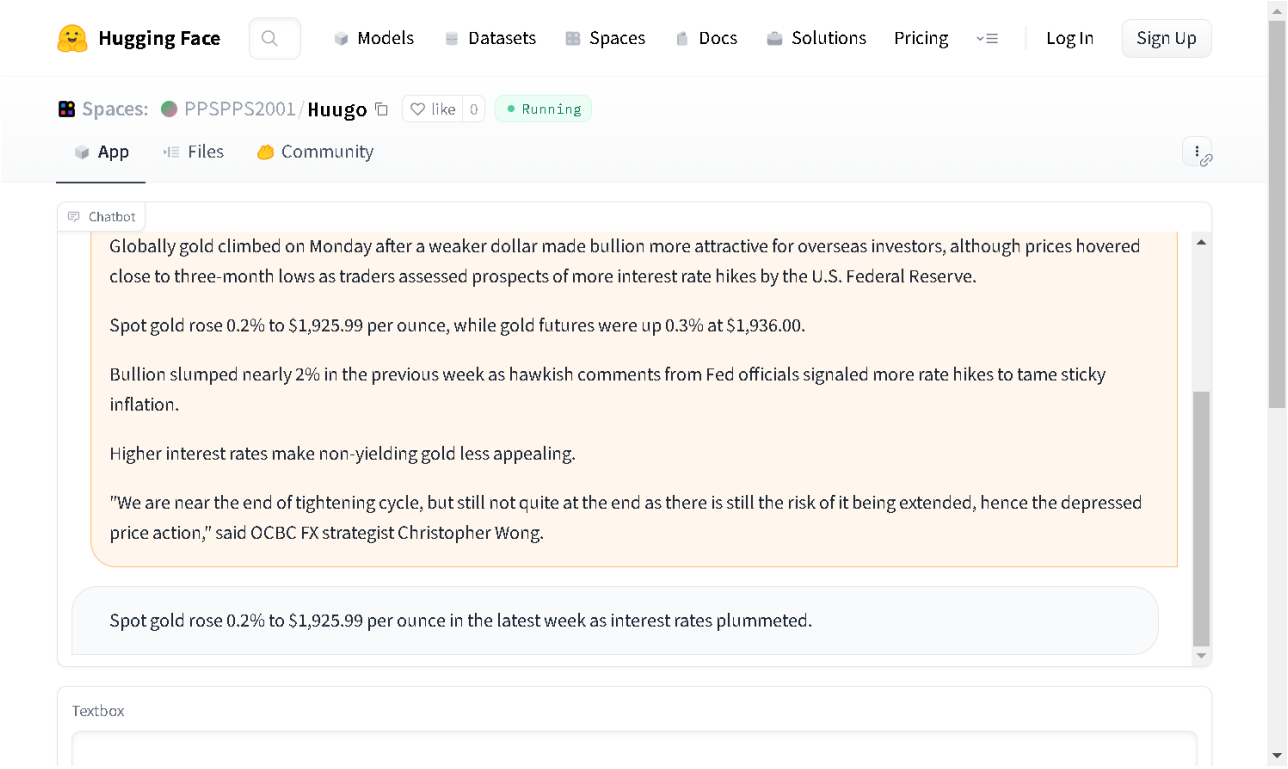


Hình 14: Chạy thử chức năng tóm tắt bản tin tiếng Anh bằng văn bản

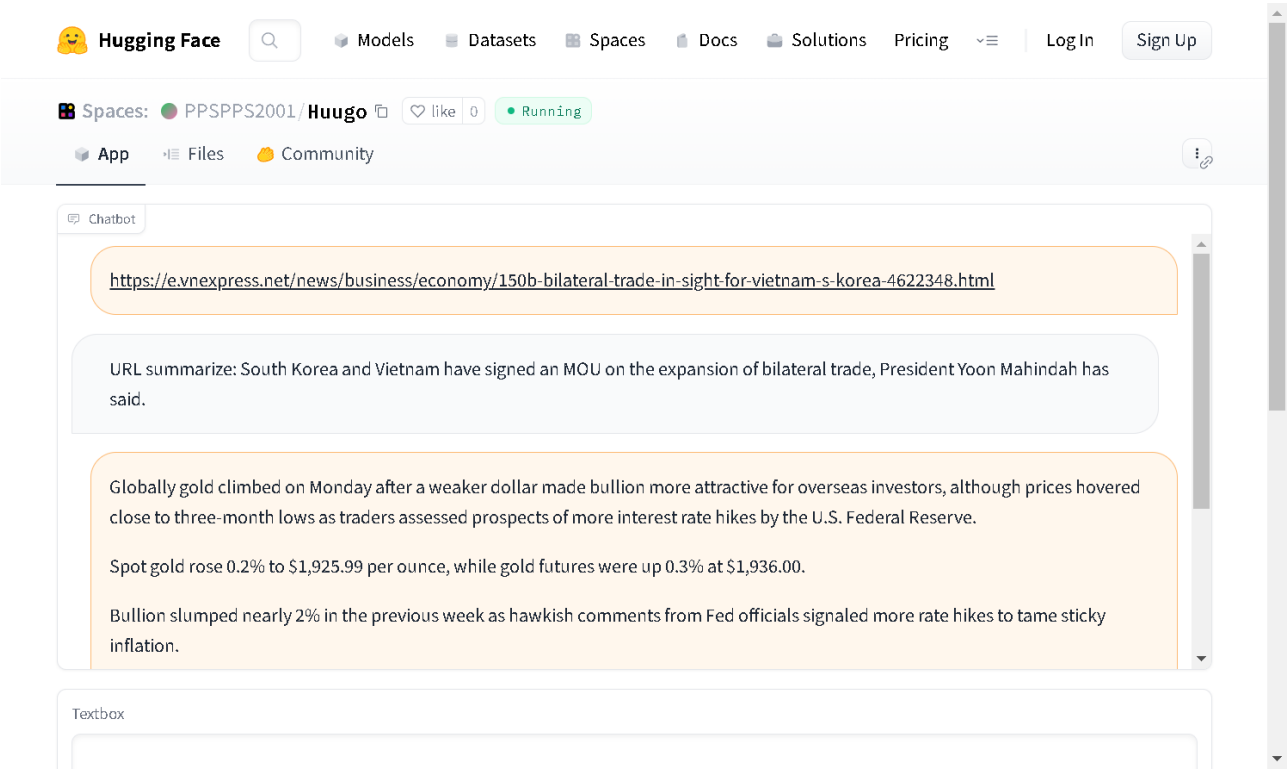


Hình 15: Chạy thử chức năng tóm tắt bản tin tiếng Anh bằng url (chỉ hỗ trợ VnExpress International)

Vì bộ weight của mô hình nhóm em không thể tải lên được trên các trang host web miễn phí. Vậy nên nhóm em đã host dưới dạng một botchat trên HuggingFace.



Hình 16: Tóm tắt bản tin tiếng Anh bằng văn bản trên HuggingFace



Hình 17: Tóm tắt bản tin tiếng Anh bằng url trên HuggingFace (chỉ hỗ trợ VnExpress International)

Tài liệu tham khảo

[1] *T5 v4.30.0*. 2023. URL: https://huggingface.co/docs/transformers/model_doc/t5.

[2] *XL-Sum dataset*. 2021. URL: https://huggingface.co/datasets/csebuetnlp/xlsum.

[3] Tahmid Hasan et al. "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703. DOI: 10.18653/v1/2021.findings-acl.413. URL: https://aclanthology.org/2021.findings-acl.413.

[4] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[5] Grzegorz Kondrak. "N-Gram Similarity and Distance". In: *String Processing and Information Retrieval*. Ed. by Mariano Consens and Gonzalo Navarro. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 115–126. ISBN: 978-3-540-32241-2.

[6] *Python ROUGE Implementation*. 2023. URL: https://github.com/google-research/google-research/tree/master/rouge.

[7] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *CoRR* abs/1910.10683 (2019). arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683.

[8] NotesOnAI. *Attention Mechanism*. https://notesonai.com/Attention+Mechanism. accessed on 2023-07-01.

[9] Alexander M Rush, Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization". In: *arXiv preprint arXiv:1509.00685* (2015).

[10] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[11] Jan A. Botha et al. "Natural Language Processing with Small Feed-Forward Networks". In: *CoRR* abs/1708.00214 (2017). arXiv: 1708.00214. URL: http://arxiv.org/abs/1708.00214.

[12] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[13] Kuzman Ganchev and Mark Dredze. "Small Statistical Models by Random Feature Mixing". In: *Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 19–20. URL: https://aclanthology.org/W08-0804.

[14] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[15] He Lyu et al. "Advances in neural information processing systems". In: *Advances in neural information processing systems* 32 (2019).

[16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[17] Takashi Matsubara. *Hugging Face Fine-Tuning Scripts for Japanese NLP*. 2021. URL: %5Curl%7Bhttps://github.com/tsmatz/huggingface-finetune-japanese%7D.