ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO
ĐỒ ÁN
Học kỳ II, năm học 2023 - 2024
Học phần:

XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Đề tài: Xây dựng mô hình tóm tắt văn bản tiếng anh

Số phách *(Do hội đồng chấm thi ghi)*

Thừa Thiên Huế, ngày tháng năm 2024

ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO
ĐỒ ÁN
Học kỳ II, năm học 2023 - 2024
Học phần:

XỬ LÝ NGÔN NGỮ TỰ NHIÊN
Đề tài: Xây dựng mô hình tóm tắt văn bản tiếng anh

Giảng viên hướng dẫn: TS. Đoàn Thị Hồng Phước
Sinh viên thực hiện: Nguyễn Thanh Lâm
Lê Thị Ngọc Thắm
Phan Thế Minh Châu
Lớp: Khoa học dữ liệu & Trí tuệ nhân tạo
Khóa: 1

Số phách
(Do hội đồng chấm thi ghi)

Thừa Thiên Huế, ngày tháng năm 2024

ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

PHIẾU ĐÁNH GIÁ ĐỒ ÁN/TIỂU LUẬN/BÀI TẬP LỚN Học kỳ II, năm học 2023 - 2024

Cán bộ chấm thi 1	Cán bộ chấm thi 2
Nhận xét: .... **Điểm đánh giá của CBCT1:** Bằng số: ..... Bằng chữ:	**Nhận xét:** .... **Điểm đánh giá của CBCT2:** Bằng số: ..... Bằng chữ: .....

Điểm kết luận: .....

Bằng số:.....

Bằng chữ: .....

Thừa Thiên Huế, ngày…tháng…năm 2024

Cán bộ chấm thi 1	Cán bộ chấm thi 2
(Ký và ghi rõ họ và tên)	*(Ký và ghi rõ họ và tên)*

MỤC LỤC

DANH MỤC HÌNH ẢNH

CHƯƠNG 1: MỞ ĐẦU

1.1. Lý do chọn đề tài

Hiện nay, tóm tắt văn bản là một trong những nhiệm vụ quan trọng và phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) do sự bùng nổ của thông tin trên mạng internet và các nguồn dữ liệu khác. Việc chọn đề tài này xuất phát từ nhu cầu cấp thiết của con người trong việc nhanh chóng nắm bắt những thông tin cốt lõi từ các tài liệu dài dòng và phức tạp. Tóm tắt văn bản không chỉ giúp tiết kiệm thời gian mà còn tăng hiệu quả trong việc truy xuất và xử lý thông tin. Hơn nữa, các ứng dụng của tóm tắt văn bản trong các hệ thống tìm kiếm, trợ lý ảo, và công cụ phân tích dữ liệu đang ngày càng phát triển, góp phần nâng cao trải nghiệm người dùng và tối ưu hóa quy trình làm việc. Do đó, nghiên cứu và phát triển các phương pháp tóm tắt văn bản tự động hiện đang là một hướng đi quan trọng và đầy triển vọng trong lĩnh vực trí tuệ nhân tạo và học máy.

1.2. Mục tiêu nghiên cứu

Phát triển một hệ thống sử dụng mô hình học máy Seq2Seq (Sequence-to-Sequence) để tóm tắt văn bản theo hướng tóm tắt trừu tượng (Abstract Summarization), nhằm tạo ra các bản tóm tắt ngắn gọn, chính xác và giữ được thông tin cốt lõi từ văn bản gốc.

1.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này là mô hình Seq2Seq (Sequence-to-Sequence) trong việc tóm tắt văn bản tự động, tập trung vào kiến trúc Encoder-Decoder và biến thể của RNN (Recurrent Neural Networks) là LSTM (Long Short-Term Memory) cùng với bộ dữ liệu các văn bản tóm tắt của những bài viết hoàn chỉnh.

1.4. Phương pháp nghiên cứu

- Thu thập dữ liệu.
- Xử lý dữ liệu.
- Xây dựng mô hình Sep2Sep.
- Huấn luyện mô hình.
- Kiểm thử và đánh giá.

1.5. Phạm vi nghiên cứu

Phạm vị nghiên cứu của đồ án xoay quanh việc xây dựng và triển khai mô hình Sep2Sep để đưa ra hệ thống tóm tắt văn bản trừu tượng từ một đoạn văn hoặc bài viết mà người sử dụng nhập vào.

1.6. Cấu trúc đồ án

Cấu trúc đồ án sẽ bao gồm các chương sau:

CHƯƠNG 1: MỞ ĐẦU

CHƯƠNG 2: KIẾN THỨC TỔNG QUAN

CHƯƠNG 3: CHỌN DATASET ĐỂ HUẤN LUYỆN VÀ KIỂM THỬ

CHƯƠNG 4: KẾT QUẢ

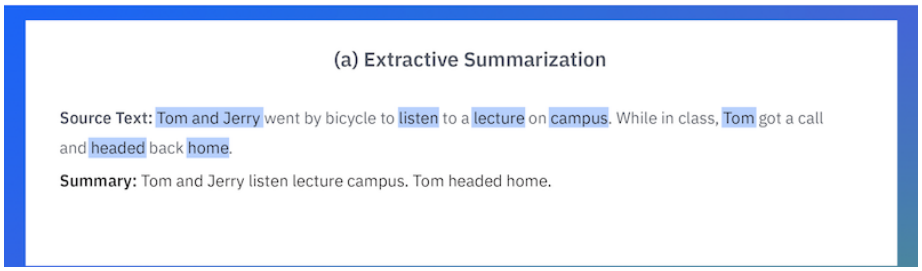
CHƯƠNG 2: KIẾN THỨC TỔNG QUAN

2.1. Tóm tắt văn bản

2.1.1. Các loại tóm tắt văn bản

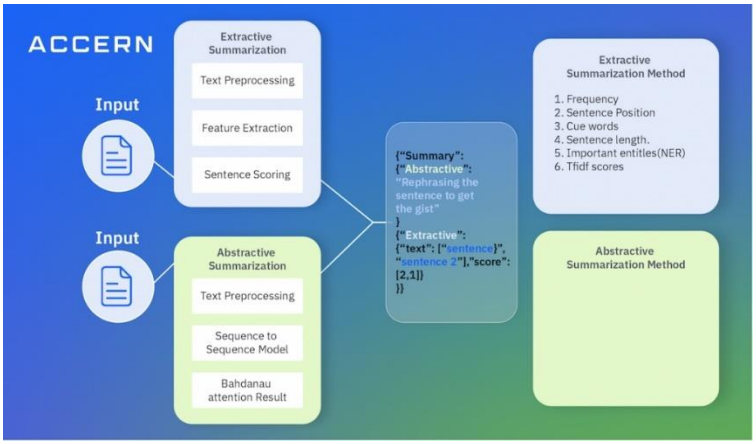
Tóm tắt văn bản là quá trình phân chia văn bản dài thành các đoạn hoặc câu ngắn gọn, dễ hiểu, đồng thời vẫn giữ được các thông tin quan trọng và ý nghĩa của văn bản gốc. Quá trình này giúp giảm thời gian cần thiết để nắm bắt nội dung của các văn bản dài như bài viết mà không làm mất đi các thông tin quan trọng. Tóm tắt văn bản có thể được thực hiện theo hai phương pháp chính[1]:

- Tóm tắt trích rút (Extract Summarization): là các tóm tắt được xây dựng bằng cách rút ra y nguyên, không thay đổi những câu chứa nội dung quan trọng trong văn bản gốc.



Hình 2.1 *Quy trình tóm tắt trích rút*

- Tóm tắt trừu tượng (Abstract Summarization): là các tóm tắt mà một số thành phần của nó không xuất hiện trong văn bản gốc mà do tác giả đưa vào, ví dụ như các câu, các thành ngữ, các chú giải…Các thuật toán tóm tắt văn bản trừu tượng tạo ra các cụm từ và câu mới chuyển tiếp thông tin hữu ích nhất từ văn bản gốc — giống như con người vẫn làm.

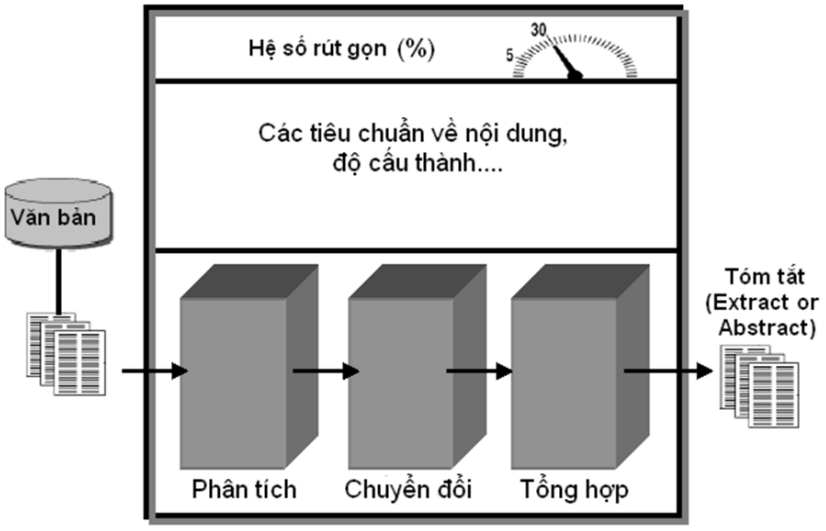


Hình 2.2 *Quy trình tóm tắt trừu tượng*

Một số nghiên cứu đã được thực hiện theo hướng xây dựng nên Tóm tắt Extract, tuy vậy hầu hết các nghiên cứu còn lại cho TTVB đều thực hiện theo hướng xây dựng Tóm tắt Abstract. Bởi vì để xây dựng một hệ thống thực hiện Tóm tắt Abstract giống như con người có thể làm, hệ thống đó không chỉ có khả năng đọc-hiểu văn bản gốc mà còn phải có khả năng tự "xây dựng văn bản" từ những từ khóa, thành ngữ, khái niệm cho trước. Một hệ thống như vậy đòi hỏi phải có cơ sở tri thức cũng như khả năng tính toán khổng lồ, khó có thể thực hiện hoàn hảo được trong hoàn cảnh hiện nay.

Tuy nhiên, đề tài này chọn theo hướng Abstract Summarization nhằm đạt được sự mạch lạc và tự nhiên trong việc diễn đạt nội dung văn bản gốc. Hệ thống Tóm tắt Abstract hướng tới việc không chỉ hiểu văn bản gốc mà còn tạo ra văn bản tóm tắt mới mạch lạc từ những khái niệm và từ khóa đã được xác định.

2.1.2. Mô hình bên ngoài của một hệ thống tóm tắt



Hình 2.3 *Mô hình bên ngoài một hệ thống Tóm tắt*

Đây là mô hình hệ thống tóm tắt được xem xét từ bên ngoài, dựa trên các đặc điểm phân loại và tiêu chí thực hiện tóm tắt. Dưới đây là mô tả tổng quan về quy trình thực hiện bên trong của một hệ thống, trong đó mô hình bên ngoài được hiểu như một quá trình bao gồm Phân tích - Chuyển đổi - Tổng hợp.

Tuy nhiên, đề tài này chọn theo hướng Abstract Summarization nhằm đạt được sự mạch lạc và tự nhiên trong việc diễn đạt nội dung văn bản gốc. Hệ thống Tóm tắt Abstract hướng tới việc không chỉ hiểu văn bản gốc mà còn tạo ra văn bản tóm tắt mới mạch lạc từ những khái niệm và từ khóa đã được xác định.

2.2. Mô hình Sequence-to-Sequence

2.2.1. Khái niệm

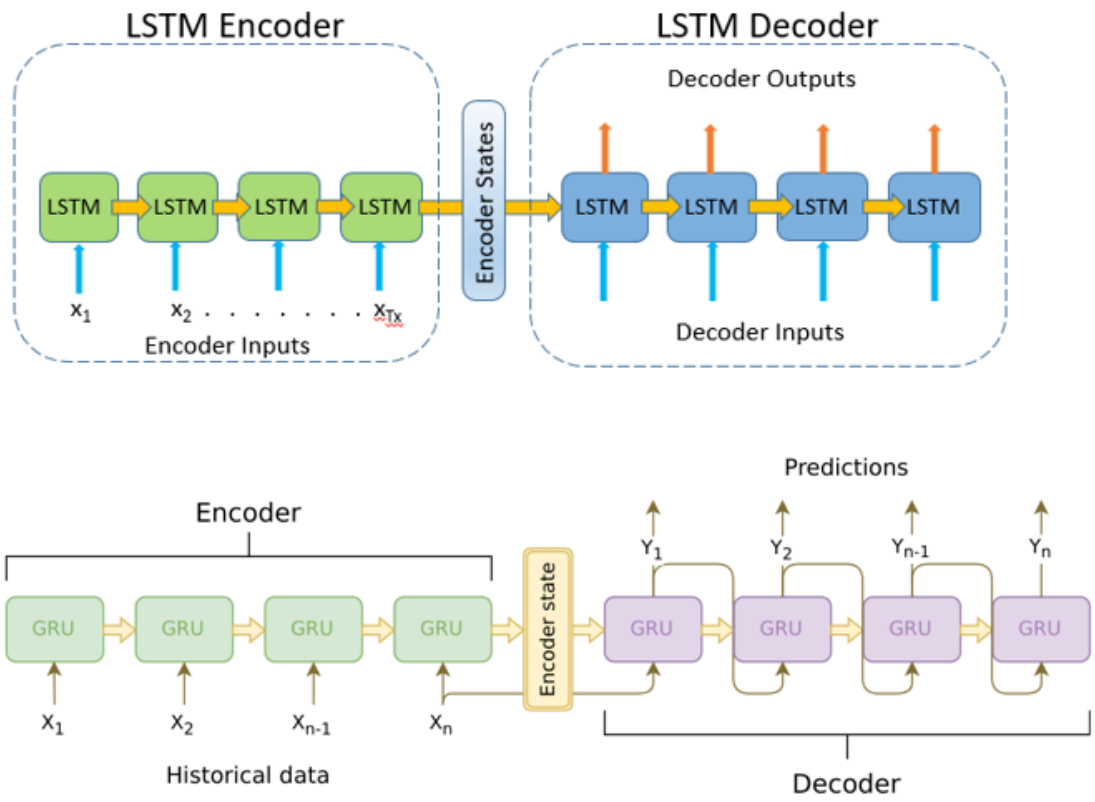
Mô hình Seq2Seq (Sequence-to-Sequence) là một dạng mạng nơ-ron nhân tạo, được thiết kế đặc biệt với kiến trúc Mạng Nơ-ron Hồi quy để chuyển đổi một dãy dữ liệu dạng chuỗi thành một dãy dữ liệu dạng chuỗi khác. Seq2Seq là khả năng xử lý linh hoạt các tác vụ mà đầu vào và đầu ra đều là chuỗi dữ liệu có độ dài bất kỳ - điều mà các mạng nơ-ron truyền thống thường gặp khó khăn [2].

Các ứng dụng tiêu biểu của Mô hình Sequence-to-Sequence (Seq2Seq):

- Dịch thuật Máy: Mô hình Seq2Seq có khả năng dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác một cách hiệu quả, ví dụ như chuyển đổi câu tiếng Anh sang tiếng Pháp hoặc tiếng Việt.
- Tóm tắt Văn bản: Seq2Seq có thể tạo ra bản tóm tắt ngắn gọn, súc tích cho các văn bản dài, chỉ tập trung vào những thông tin quan trọng và lược bỏ những chi tiết phụ.
- Nhận dạng Giọng nói: Mô hình này có thể chuyển đổi âm thanh thành văn bản. Seq2Seq được huấn luyện để biến đổi tín hiệu âm thanh (dòng âm thanh) thành các bản ghi chép tương ứng (dòng chữ).
- Chú thích Hình ảnh: Seq2Seq có thể mô tả nội dung hình ảnh bằng ngôn ngữ tự nhiên. Bộ mã hóa sẽ xử lý hình ảnh (thường sử dụng Mạng nơ-ron tích chập, CNN) để tạo ra một vector ngữ cảnh, sau đó bộ giải mã sẽ biến đổi vector này thành một câu mô tả chi tiết.
- Chú thích Video: Tương tự như chú thích hình ảnh, Seq2Seq có thể tạo ra các chú thích văn bản cho video, tóm tắt nội dung video, bao gồm cả chuỗi hành động và cảnh quay.

Mục tiêu của bài đồ án này là xây dựng một chương trình tóm tắt văn bản với đầu vào là một chuỗi dài các từ (trong nội dung văn bản) và đầu ra là một bản tóm tắt ngắn (cũng là một chuỗi) theo dạng tóm tắt trừu tượng (Abstract Summarization).

2.2.2. Kiến trúc



Hình 2.4. *Kiến trúc LSTM&GRU Seq2seq*

Mô hình Seq2seq sử dụng kiến trúc Encoder-Decoder có độ dài đầu vào và đầu ra khác nhau. Kiến trúc Encoder-Decoder được coi là hai khối - Mã hóa (Encoder) và Giải mã (Decoder), hai khối này được kết nối với nhau thông qua Vector trung gian (Context Vector)[2]:

- Bộ mã hóa - Encoder: Bộ mã hóa thường sử dụng kiến trúc mạng Long Short-Term Memory (LSTM) hoặc GRU. Encoder đọc toàn bộ chuỗi dữ liệu đầu vào, xử lý từng token trong chuỗi đầu vào đó, và nó cố gắng nhồi nhét toàn bộ thông tin đầu vào vào một vector có độ dài cố định, tức là "vector trung gian". Sau đó bộ mã hóa sẽ chuyển vector này sang bộ giải mã.
- Vector trung gian - Context Vector: Vector này có chức năng gói gọn toàn bộ ý nghĩa của chuỗi đầu vào và giúp bộ giải mã đưa ra được quyết định chính xác.

- Đây là trạng thái ẩn nằm cuối chuỗi và được tính bởi bộ mã hóa, vector này sau đó cũng hoạt động như trạng thái ẩn đầu tiên của bộ giải mã.
- Bộ giải mã - Decoder: Bộ giả mã cũng sử dụng kiến trúc LSTM (hoặc GRU). Trạng thái đầu vào của Decoder được khởi tạo dựa trên trạng thái cuối cùng của Encoder. Nói cách khác, vector trung gian ô nhớ cuối cùng của Encoder sẽ được đưa vào ô nhớ đầu tiên của mạng Decoder. Bằng cách sử dụng các trạng thái khởi tạo này, Decoder bắt đầu tạo ra chuỗi đầu ra. Đồng thời, các đầu ra này cũng được sử dụng để dự đoán các đầu ra tiếp theo.

2.2.3. Ưu điểm

Khả năng xử lý linh hoạt: Điểm nổi bật của Seq2Seq là khả năng xử lý linh hoạt các chuỗi dữ liệu đầu vào và đầu ra có độ dài bất kỳ, nhờ cấu trúc encoder-decoder thông minh. Nhờ vậy, Seq2Seq trở thành công cụ đắc lực cho các tác vụ như dịch máy, nơi độ dài câu đầu vào và đầu ra có thể khác biệt đáng kể.

Nắm bắt ngữ cảnh hiệu quả: Sử dụng các mạng nơ-ron hồi quy (RNN) như Long Short-Term Memory (LSTM) và GRU trong cấu trúc encoder-decoder, Seq2Seq có khả năng nắm bắt các mối phụ thuộc xa trong chuỗi đầu vào. Đây là yếu tố then chốt để hiểu ngữ cảnh và ý nghĩa trong các tác vụ như tóm tắt văn bản và dịch máy neuron.

Nâng cao hiệu suất với Attention Mechanism: Cơ chế Attention (chú ý) ra đời giúp nâng cao hiệu suất của Seq2Seq bằng cách cho phép bộ giải mã tập trung vào các phần quan trọng của chuỗi đầu vào tại mỗi bước xử lý. Cách tiếp cận này khắc phục hạn chế của việc nén tất cả thông tin đầu vào thành một vector ngữ cảnh duy nhất, đồng thời cải thiện đáng kể độ chính xác trong các tác vụ đòi hỏi sự hiểu biết sâu sắc như chú thích hình ảnh và xử lý ngôn ngữ tự nhiên (NLP).

Đa dạng ứng dụng: Không chỉ giới hạn trong xử lý văn bản, Seq2Seq còn được ứng dụng hiệu quả trong nhận dạng giọng nói, chú thích video và dự báo chuỗi thời gian. Khả năng xử lý và tạo chuỗi của Seq2Seq biến nó thành công cụ mạnh mẽ trong nhiều ứng dụng học sâu khác nhau [2].

2.2.4. Nhược điểm

Độ phức tạp tính toán: Việc huấn luyện mô hình Seq2Seq có thể đòi hỏi nhiều tài nguyên tính toán, đặc biệt là đối với các mô hình sử dụng mạng Long Short-Term Memory (LSTM) hoặc GRU.

Khó khăn trong xử lý chuỗi dài: Mặc dù các mạng nơ-ron hồi quy (RNN) như LSTM và GRU được thiết kế để xử lý dữ liệu chuỗi, chúng có thể gặp khó khăn với các chuỗi dài do vấn đề vanishing gradient (mất gradient). Vấn đề này ảnh hưởng đến việc học các mối phụ thuộc xa trong chuỗi.

Phụ thuộc vào bộ dữ liệu lớn: Mô hình Seq2Seq cần bộ dữ liệu phong phú và đa dạng để huấn luyện hiệu quả. Tuy nhiên, sự phụ thuộc này có thể dẫn đến hai mặt:

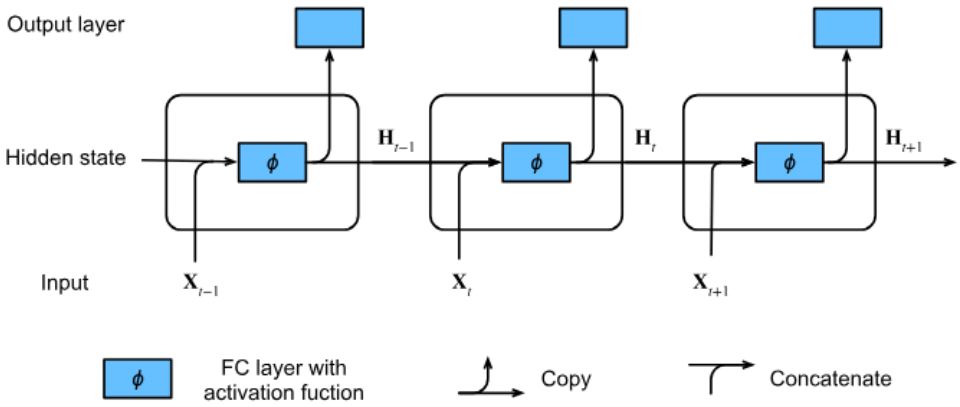
- Thiếu dữ liệu huấn luyện: Dữ liệu huấn luyện không đủ hoặc chất lượng kém có thể dẫn đến overfitting (hiết khớp quá mức).
- Dữ liệu sai lệch: Dữ liệu huấn luyện sai lệch có thể khiến mô hình học sai mối quan hệ giữa các phần tử trong chuỗi, dẫn đến kết quả dự đoán không chính xác.

Hiệu suất của mô hình Seq2Seq có thể thay đổi đáng kể tùy thuộc vào lựa chọn kiến trúc và siêu tham số, chẳng hạn như số lượng lớp trong encoder-decoder, kích thước trạng thái ẩn và thuật toán tối ưu hóa được sử dụng.

Sự cạnh tranh từ Transformers: Transformers và các biến thể của nó (như BERT và GPT) đã chứng minh khả năng vượt trội so với các mô hình Seq2Seq truyền thống trong nhiều tác vụ bằng cách loại bỏ sự cần thiết cho xử lý tuần tự và xử lý tốt hơn các mối phụ thuộc xa[2].

2.3. Mạng nơ-ron hồi quy (RNN)

2.3.1. Khái niệm



Hình 2.5. *Mạng nơ-ron hồi quy (RNN)*

Mạng nơ-ron hồi quy (RNN) là một mô hình học sâu được đào tạo để xử lý và chuyển đổi đầu vào dữ liệu tuần tự thành đầu ra dữ liệu tuần tự cụ thể. Dữ liệu tuần tự là dữ liệu, chẳng hạn như từ, câu hoặc dữ liệu chuỗi thời gian, trong đó các thành phần tuần tự tương quan với nhau dựa trên ngữ nghĩa phức tạp và quy tắc cú pháp. RNN là một hệ thống phần mềm gồm nhiều thành phần được kết nối với nhau theo cách con người thực hiện chuyển đổi dữ liệu tuần tự, chẳng hạn như dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. Phần lớn RNN đang được thay thế bằng trí tuệ nhân tạo (AI) dựa trên công cụ biến đổi và các mô hình ngôn ngữ lớn (LLM), hiệu quả hơn nhiều trong việc xử lý dữ liệu tuần tự[3].

2.3.2. Phân loại

Có bốn loại mạng nơ-ron hồi quy (RNN) được phân loại dựa trên số lượng đầu vào và đầu ra của mạng:

One to One: Loại RNN này hoạt động giống như một mạng nơ-ron thông thường, cũng được gọi là Mạng Nơ-ron Vanilla. Trong mạng này, chỉ có một đầu vào và một đầu ra.

One To Many: Trong loại RNN này, mạng nhận một đầu vào và tạo ra nhiều đầu ra liên quan đến đầu vào đó. Một ví dụ điển hình của dạng mạng này chính là chú thích ảnh (Image Captioning). Ở tác vụ này, dựa trên một hình ảnh đầu vào, mạng sẽ dự đoán một câu chứa nhiều từ để mô tả nội dung của bức ảnh.

Many to One: Trong loại mạng này, nhiều đầu vào được đưa vào mạng ở nhiều trạng thái khác nhau của mạng, tạo ra duy nhất một đầu ra. Loại mạng này được sử dụng trong các bài toán như phân tích cảm xúc, nơi chúng ta đưa vào nhiều từ làm đầu vào và chỉ dự đoán cảm xúc của câu làm đầu ra.

Many to Many: Trong loại mạng nơ-ron này, có nhiều đầu vào và nhiều đầu ra tương ứng với một vấn đề. Một ví dụ về vấn đề này là dịch ngôn ngữ. Trong dịch ngôn ngữ, chúng ta cung cấp nhiều từ từ một ngôn ngữ làm đầu vào và dự đoán nhiều từ từ ngôn ngữ thứ hai làm đầu ra [3].

2.3.3. Ưu và nhược điểm

Ưu điểm:

- Có khả năng xử lý dữ liệu đầu vào ở bất kỳ độ dài nào.
- Kích thước mô hình không tăng theo kích thước đầu vào.
- Việc huấn luyện mô hình có sử dụng thông tin ở Time-Step trước đó.
- Các hệ số của mô hình (weight và bias) được chia sẻ theo thời gian.

Nhược điểm:

- Việc xử lý, tính toán mất khá nhiều thời gian.
- Thông tin từ các Time-Step ở xa không được duy trì tốt.
- Không thể xem xét bất kỳ đầu vào nào trong tương lai cho trạng thái hiện tại.

2.4. Điểm Rouge

2.4.1. Khái niệm

Điểm ROUGE, hay Recall-Oriented Understudy for Gisting Evaluation, là một tập hợp các chỉ số được sử dụng để đánh giá chất lượng của các mô hình dịch và tóm tắt tài liệu. Nó đo lường sự trùng lặp giữa tóm tắt hoặc bản dịch do hệ thống tạo ra và một tập hợp các tóm tắt hoặc bản dịch tham chiếu do con người tạo ra, sử dụng các kỹ thuật khác nhau như thống kê đồng xuất hiện n-gram, tỷ lệ trùng lặp từ, và các chỉ số tương đồng khác. Điểm số dao động từ 0 đến 1, với điểm số gần bằng không chỉ ra sự tương đồng kém giữa ứng cử viên và các tham chiếu, và điểm số gần bằng một chỉ ra sự tương đồng mạnh mẽ.

Điểm ROUGE cao hơn chỉ ra hiệu suất tốt hơn trong việc bảo tồn thông tin chính từ văn bản gốc trong khi tạo ra một tóm tắt hoặc bản dịch ngắn gọn.

Điểm ROUGE dựa trên khái niệm n-gram, là các chuỗi gồm n từ. Các loại chỉ số ROUGE khác nhau bao gồm:

- ROUGE-N.
- ROUGE-L.
- ROUGE-W.
- ROUGE-S.
- ROUGE-SU.

2.4.2. ROUGE–N

ROUGE-N đo lường sự trùng lặp của các n-gram (các chuỗi liên tiếp gồm n từ) giữa văn bản ứng viên và văn bản tham chiếu. Nó tính toán độ chính xác, độ bao phủ và điểm F1 dựa trên sự trùng lặp của n-gram.

Ví dụ, ROUGE-1 (unigram) đo lường sự trùng lặp của các từ đơn, ROUGE-2 (bigram) đo lường sự trùng lặp của các chuỗi hai từ, và tiếp tục như vậy.

ROUGE-N thường được sử dụng để đánh giá tính chính xác ngữ pháp và sự trôi chảy của văn bản được tạo ra.

Số điểm ROUGE-N của một bản tóm tắt được xác định như sau:

$$ROUGE - n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} Count_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} Count(gram_n)}$$

Hình 2.6. *ROUGE-N*

Trong đó:

- $Count_{match}(gram_n)$ là số lượng lớn nhất có trong kết quả tóm tắt và bản tóm tắt tham khảo.
- $Count(gram_n)$ là số lượng n-grams có trong bản tóm tắt tham khảo [4].

2.4.3. ROUGE–L

ROUGE-L đo lường dãy con chung dài nhất (LCS) giữa văn bản ứng viên và văn bản tham chiếu. Nó tính toán độ chính xác, độ bao phủ và điểm F1 dựa trên độ dài của LCS.

ROUGE-L thường được sử dụng để đánh giá sự tương đồng ngữ nghĩa và phạm vi nội dung của văn bản được tạo ra, vì nó xem xét dãy con chung bất kể thứ tự từ.

$$\begin{cases} P_{LCS}(R,S) = \dfrac{LCS(R,S)}{|S|} \\ R_{LCS}(R,S) = \dfrac{LCS(R,S)}{|R|} \\ R_{LCS}(R,S) = \dfrac{(1+\beta^2)P_{LCS}(R,S)R_{LCS}(R,S)}{\beta^2 P_{LCS}(R,S) + R_{LCS}(R,S)} \end{cases}$$

Hình 2.7. *ROUGE – L*

Trong đó:

- $|R|$ và $|S|$ tương ứng là chiều dài văn bản dẫn xuất R và văn bản ứng viên S.
- LCR(R,S) là LCS giữa R và S.
- PLCS(R,S) là độ chính xác của LCS(R,S).
- RLCS(R,S) là độ phủ của LCS(R,S).
- β là PLCS(R,S) / RLCS(R,S) [4]

2.4.4. Đặc điểm

Đánh giá dựa trên sự hồi tưởng: ROUGE tập trung vào đo lường sự hồi tưởng, đảm bảo thông tin quan trọng từ văn bản gốc không bị mất.

Linh hoạt và điều chỉnh: Có thể tuỳ chỉnh ROUGE với nhiều loại n-gram và phương pháp đo tương đồng khác nhau.

Hỗ trợ nhiều tham chiếu: ROUGE xử lý nhiều tóm tắt hoặc dịch tham chiếu, cung cấp đánh giá toàn diện hơn.

Tích hợp với các công cụ đánh giá phổ biến: ROUGE tích hợp dễ dàng với các khung công cụ xử lý ngôn ngữ tự nhiên.

Mã nguồn mở: ROUGE có sẵn dưới dạng mã nguồn mở, miễn phí cho các nhà nghiên cứu và phát triển [4].

2.4.5. Hoạt động

Tiền xử lý: Loại bỏ thông tin không cần thiết từ tóm tắt hoặc bản dịch.

Trích xuất đặc trưng: Lấy các đặc trưng quan trọng từ cả tóm tắt được tạo ra và các tóm tắt tham chiếu.

Tính điểm tương đồng: So sánh các đặc trưng từ hai văn bản, dùng các kỹ thuật như thống kê đồng xuất hiện n-gram.

Tổng hợp điểm tương đồng: Kết hợp các điểm tương đồng để tạo ra một điểm ROUGE đại diện cho hiệu suất tổng thể.

Chuẩn hóa và giải thích: Điểm ROUGE thường được chuẩn hóa từ 0 đến 1, chỉ ra hiệu suất trong việc bảo tồn thông tin chính và tạo ra một tóm tắt hoặc bản dịch ngắn gọn [4].

2.4.6. Ưu và nhược điểm

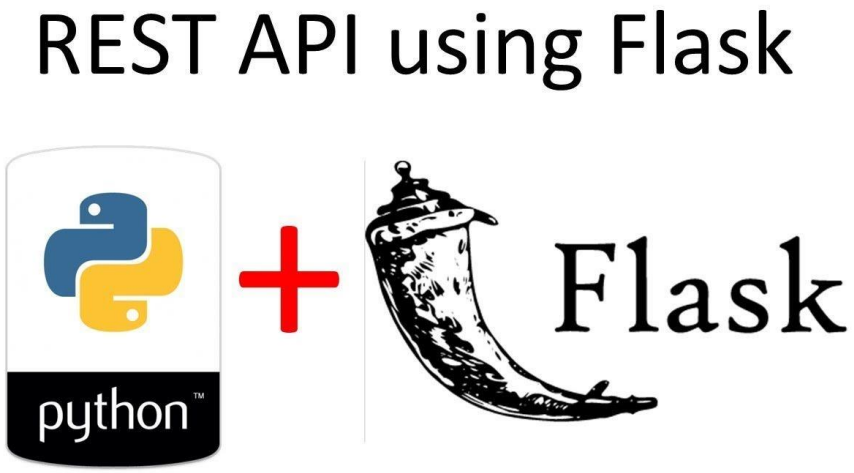
Ưu điểm:

- Độ tin cậy và tính nhất quán: ROUGE là một chỉ số chuẩn hóa được rộng rãi chấp nhận để so sánh kết quả giữa các nghiên cứu khác nhau.
- Linh hoạt và có thể điều chỉnh: ROUGE có thể tuỳ chỉnh với nhiều loại n-gram và phương pháp đo tương đồng khác nhau.
- Hỗ trợ nhiều tham chiếu: ROUGE có thể đánh giá dựa trên nhiều tóm tắt hoặc bản dịch tham chiếu.
- Tích hợp với các công cụ đánh giá phổ biến: ROUGE được hỗ trợ bởi nhiều công cụ xử lý ngôn ngữ tự nhiên.
- Mã nguồn mở: ROUGE có sẵn dưới dạng mã nguồn mở, miễn phí cho các nhà nghiên cứu và phát triển.

Nhược điểm:

- Phụ thuộc vào tóm tắt tham chiếu: ROUGE chủ yếu đo lường sự hồi tưởng và phụ thuộc vào chất lượng của tóm tắt tham chiếu.
- Nhạy cảm với kỹ thuật tiền xử lý: ROUGE có thể bị ảnh hưởng bởi cách tiền xử lý dữ liệu.
- Thiếu nhận thức về ngữ cảnh: ROUGE không xem xét bối cảnh rộng lớn của văn bản, có thể hạn chế ứng dụng của nó.
- Không thể hiện sự tương đồng ngữ nghĩa: ROUGE chủ yếu tập trung vào trùng lặp từ vựng.
- Phụ thuộc vào tóm tắt tham chiếu của con người: ROUGE phụ thuộc vào chất lượng của các tóm tắt tham chiếu [4].

2.5. Flask



Hình 2.8. *Flask*

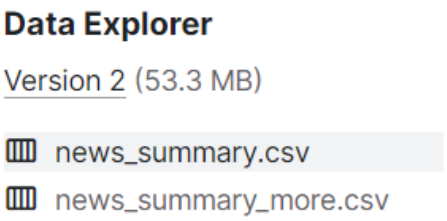
Flask là loại framework web phổ biến được viết bằng trình lập ngôn ngữ Python. Công nghệ thường được sử dụng để xây dựng trang web từ những ứng dụng đơn giản đến những hệ thống phức tạp hơn.

Flask được thiết kế để hoạt động và mở rộng một cách, đồng thời nó cũng cung cấp các công cụ và thư viện cần thiết để phát triển ứng dụng web hiệu quả. Flask cũng có cộng đồng sáng tạo và hỗ trợ mạnh mẽ từ cộng đồng Python.[5]

CHƯƠNG 3: CHỌN DATASET ĐỂ HUẤN LUYỆN VÀ KIỂM THỬ

3.1. Chuẩn bị dữ liệu

Dữ liệu được lấy từ nền tảng Kaggle bao gồm 2 tệp chính là "news_summary.csv" và "news_summary_more.csv".



Link: https://www.kaggle.com/datasets/sunnysai12345/news-summary

Bộ dữ liệu được thu thập từ các tin tức tóm tắt từ Inshorts và chỉ lấy các bài báo từ Hindu, Indian Times và Guardian. Thời gian lấy dữ liệu từ tháng 2 đến tháng 8 năm 2017.

Tập dữ liệu "news_summary.csv" bao gồm 4515 bài báo. Mỗi bài báo trong tập dữ liệu ta sử dụng hai thông tin “text” và “ctext” với text là đoạn văn bản dài còn ctext là phần tóm tắt văn bản tương ứng.

Tập dữ liệu "news_summary_more.csv" bao gồm 98280 bài báo. Mỗi bài báo trong tập dữ liệu ta sử dụng hai thông tin “text” và “headlines” với text là đoạn văn bản dài còn headlines là phần tóm tắt văn bản tương ứng.

Kết hợp hai tập dữ liệu trên ta có được tập dữ liệu lớn 102914 điểm dữ liệu gồm text và summary tương ứng.

	text	summary
0	Saurav Kant, an alumnus of upGrad and IIIT-B's...	upGrad learner switches to career in ML & Al w...
1	Kunal Shah's credit card bill payment platform...	Delhi techie wins free food from Swiggy for on...
2	New Zealand defeated India by 8 wickets in the...	New Zealand end Rohit Sharma-led India's 12-ma...
3	With Aegon Life iTerm Insurance plan, customer...	Aegon life iTerm insurance plan helps customer...
4	Speaking about the sexual harassment allegatio...	Have known Hirani for yrs, what if MeToo claim...
...	...	...
102910	Mansha Mahajan 24 Feb 2017,Friday http://india...	Rasna seeking ?250 cr revenue from snack categ...
102911	Dishant Sharma 03 Aug 2017,Thursday http://ind...	Sachin attends Rajya Sabha after questions on ...
102912	Tanya Dhingra 03 Aug 2017,Thursday http://www....	Shouldn't rob their childhood: Aamir on kids r...
102913	Pragya Swastik 07 Dec 2016,Wednesday http://in...	Asha Bhosle gets ?53,000 power bill for unused...
102914	Chhavi Tyagi 03 Aug 2017,Thursday http://india...	More than half of India's languages may die in...

102915 rows × 2 columns

Hình 3.1. *Dataset*

3.2. Huấn luyện và kiểm thử

3.2.1. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một bước cực kỳ quan trọng trước khi đưa vào xây dựng mô hình tóm tắt văn bản vì nhiều lý do sau:

Loại bỏ nhiễu: Dữ liệu thô thường chứa nhiều nhiễu như ký tự đặc biệt, dấu câu, HTML tags, và các ký tự không mong muốn khác. Việc loại bỏ những yếu tố này giúp mô hình tập trung vào nội dung chính xác của văn bản.

Thống nhất định dạng: Văn bản có thể xuất hiện dưới nhiều định dạng khác nhau. Tiền xử lý giúp chuẩn hóa định dạng văn bản, làm cho dữ liệu nhất quán và dễ dàng xử lý hơn.

Loại bỏ ký tự đặc biệt: Các ký tự tab (\t), ký tự xuống dòng (\r), (\n) thường xuất hiện trong văn bản, nhưng chúng có thể gây nhiễu trong quá trình xử lý văn bản. Thay thế chúng bằng khoảng trắng giúp làm sạch dữ liệu và chuẩn hóa khoảng cách giữa các từ.

Chuẩn hóa văn bản: Chuyển đổi toàn bộ văn bản thành chữ thường giúp giảm sự đa dạng về hình thức của các từ, giúp mô hình học máy xử lý dữ liệu hiệu quả hơn.

```
row = re.sub("(\\t)", " ", str(row)).lower()
row = re.sub("(\\r)", " ", str(row)).lower()
row = re.sub("(\\n)", " ", str(row)).lower()
```

Thay thế các chuỗi ký tự đặc biệt liên tục bằng khoảng trắng. Cụ thể, nó xử lý các chuỗi ký tự gạch dưới (__), gạch ngang (--), dấu ngã (~~), dấu cộng (++), và dấu chấm (..) lặp lại nhiều lần.

```
row = re.sub("(__+)", " ", str(row)).lower()
row = re.sub("(--+)", " ", str(row)).lower()
row = re.sub("(~~+)", " ", str(row)).lower()
row = re.sub("(\+\++)", " ", str(row)).lower()
row = re.sub("(\.\.+)", " ", str(row)).lower()
```

Thay thế các ký tự đặc biệt khác nhau bằng khoảng trắng.

```
row = re.sub(r"[<>()|&©ø\[\]\'\",;?~*!]", " ", str(row)).lower()
```

Thay thế các chuỗi ký tự đặc biệt theo các mẫu nhất định bằng các chuỗi chung như "INC_NUM" và "CM_NUM". Điều này giúp làm cho dữ liệu nhất quán và dễ dàng xử lý hơn.

```
        row = re.sub("([iI][nN][cC]\d+)", "INC_NUM", str(row)).lower()
        row = re.sub("([cC][mM]\d+)|([cC][hH][gG]\d+)", "CM_NUM",
str(row)).lower()
```

Loại bỏ dấu câu không cần thiết: Dấu chấm, dấu gạch ngang, và dấu hai chấm khi xuất hiện ở cuối từ và ngay trước khoảng trắng thường không mang thông tin ngữ nghĩa quan trọng và có thể gây nhiễu trong quá trình xử lý văn bản.

```
        row = re.sub("(\.\s+)", " ", str(row)).lower()
        row = re.sub("(\-\s+)", " ", str(row)).lower()
        row = re.sub("(\:\s+)", " ", str(row)).lower()
```

Các URL đầy đủ thường không mang nhiều ý nghĩa ngữ nghĩa và có thể gây nhiễu cho các mô hình học máy. Bằng cách thay thế chúng bằng tên miền, văn bản trở nên gọn gàng hơn và tập trung vào thông tin quan trọng.

```
  try:
            url = re.search(r"((https*:\/*)([^\/\s]+))(.[^\s]+)",
str(row))
            repl_url = url.group(3)
            row = re.sub(r"((https*:\/*)([^\/\s]+))(.[^\s]+)",
repl_url, str(row))
        except:
            pass
```

Các stopwords là những từ phổ biến trong một ngôn ngữ nhưng thường không mang nhiều ý nghĩa ngữ nghĩa trong quá trình phân tích văn bản, như "the", "and", "is", "in",...

Loại bỏ stopwords từ văn bản giúp làm sạch và tinh chỉnh dữ liệu văn bản trước khi tiến hành các phân tích ngôn ngữ tự nhiên hoặc xây dựng các mô hình học máy. Điều này giúp tăng hiệu suất của các phương pháp phân tích ngôn ngữ và đảm bảo rằng các từ quan trọng hơn được tập trung.

```
def rm_stopwords_from_text(text):
    _stopwords = stopwords.words('english')
    text = text.split()
    word_list = [word for word in text if word not in _stopwords]
    return ' '.join(word_list)
```

Xử lý các contraction.

```
#liệt kê các contraction để chút xử lý
contraction_mapping = {"ain't": "is not", "aren't": "are not","can't":
                       "didn't": "did not", "doesn't": "does not", "do
                       "he'd": "he would","he'll": "he will", "he's":
                       "I'd": "I would", "I'd've": "I would have", "I'
```

```
      row = ' '.join([contraction_mapping[t] if t in contraction_mapping
else t for t in row.split(" ")])
```

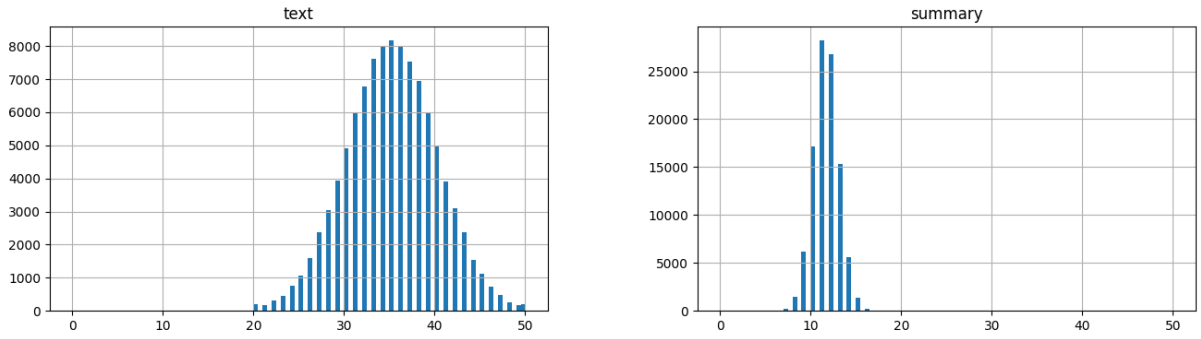
Sử dụng các token bắt ”_START_” và ”_END_” trong văn bản tóm tắt là một phương pháp quan trọng trong xử lý văn bản, đặc biệt khi xây dựng các mô hình tóm tắt hoặc dịch máy. Cách tiếp cận này giúp mô hình hiểu được vị trí bắt đầu và kết thúc của tóm tắt trong văn bản nguồn, từ đó hỗ trợ quá trình học một cách hiệu quả hơn.

```
      summary = ['_START_ '+ str(doc) + ' _END_' for doc in
nlp.pipe(processed_summary, batch_size=5000)]
```

Việc sử dụng các token này sẽ bị trình phân tích của TensorFlow có thể lọc và chuyển đổi các token sang chữ thường, khi _START_ và _END_ sẽ bị chuyển đổi thành start và end. Do đó, ta sử dụng sostok và eostok là giải pháp hiệu quả hơn để xác định vị trí bắt đầu và kết thúc của tóm tắt.

```
      post_pre['summary'] = post_pre['summary'].apply(lambda x: 'sostok
' + x \ + ' eostok')
```

Nếu độ dài của tiêu đề hoặc văn bản được giữ ở mức lớn thì mô hình học sâu sẽ gặp phải các vấn đề về hiệu suất và quá trình đào tạo cũng sẽ chậm hơn. Vì vậy cần xác định độ dài tối đa của văn bản và đoạn tóm tắt.



Hình 3.2. *Biểu đồ biểu diễn sộ lượng từ trong text*

Dựa vào biểu đồ Hình 3.2 ta có thể lựa chọn max_text_len cho văn bản là 42 và max_summary_len cho đoạn tóm tắt là 13.

Tiếp đến là chia tập dữ liệu thành train và val (validation) là một bước quan trọng trong quá trình học máy, bao gồm cả việc huấn luyện mô hình tóm tắt văn bản. Việc

chia tập dữ liệu giúp hiệu suất mô hình được đánh giá dựa trên khả năng tóm tắt chính xác các văn bản trong tập val. Việc sử dụng tập val giúp đảm bảo rằng mô hình không chỉ học thuộc lòng dữ liệu train mà còn có thể tổng hợp tốt các văn bản mới. Ngoài ra còn giúp ngăn chặn Overfitting và điều chỉnh mô hình để đạt được kết quả tốt nhất.

Chúng ta sử dụng hàm train_test_split của thư viện sklearn để chia dữ liệu thành tập train và val theo tỷ lệ 90-10.

```
x_tr, x_val, y_tr, y_val = train_test_split(
            np.array(post_pre["text"]),
            np.array(post_pre["summary"]),
            test_size=0.1,
            random_state=0,
            shuffle=True,)
```

3.2.2. Word Embedding

Word embedding là kỹ thuật biểu diễn các từ trong ngôn ngữ tự nhiên (NLP) thành các vectơ số thực. Mỗi vectơ đại diện cho ngữ nghĩa của từ đó và các mối quan hệ giữa các từ. Word embedding giúp mô hình NLP hiểu được ý nghĩa của văn bản một cách tốt hơn. Có nhiều phương pháp để thực hiện công việc chuyển văn bản thành vector, trong bài báo cáo này chúng em sử dụng công cụ Tokenizer của thư viện TensorFlow.keras.preprocessing để chuyển các đoạn text thành vector.

Để chuyển các câu văn thành ma trận số, chúng ta cần tạo ra một từ điển mapping mỗi từ với index tương ứng của nó. module tokenizer dễ dàng giúp ta thực hiện việc này.

```
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_tr))
y_tokenizer = Tokenizer()
y_tokenizer.fit_on_texts(list(y_tr))
```

Sau đó dựa vào tập từ điển có thể chuyển tập chứa văn bản sang vector index tương ứng.

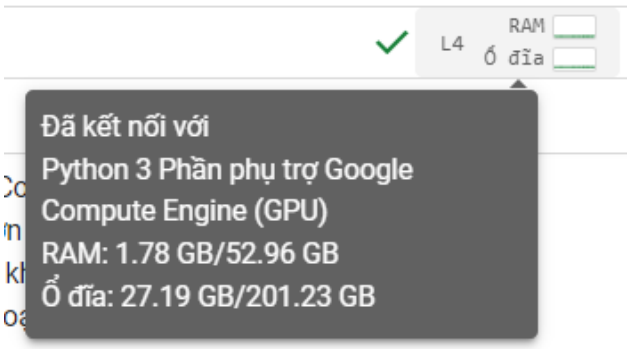
```
x_tr_seq = x_tokenizer.texts_to_sequences(x_tr)
x_val_seq = x_tokenizer.texts_to_sequences(x_val)
y_tr_seq = y_tokenizer.texts_to_sequences(y_tr)
y_val_seq = y_tokenizer.texts_to_sequences(y_val)
```

Nếu đoạn text bé hơn max_summary_len thì điền 0 vào cho đủ, đoạn text lớn hơn max_summary_len thì cắt bớt các chữ ở cuối cho đủ.

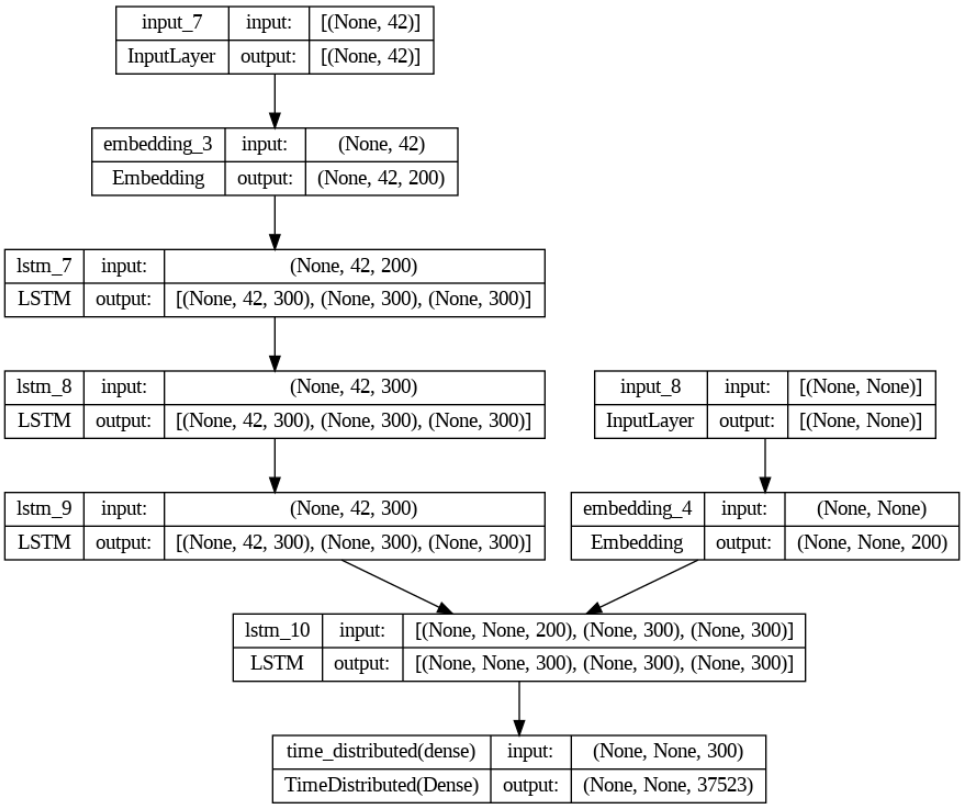
```
y_tr = pad_sequences(y_tr_seq, maxlen=max_summary_len, padding='post')
y_val = pad_sequences(y_val_seq, maxlen=max_summary_len,padding='post')
```

3.2.3. Xây dựng mô hình

Train mô hình sử dụng L4 GPU của google colab.



Xác định kiến trúc của mô hình:



Hình 3.3. *Kiến trúc mô hình*

Từ Hình 3.3 thấy được:

- Lớp đầu vào: Lớp này nhận đầu vào là một chuỗi các token. Mỗi token là một số nguyên đại diện cho một từ hoặc cụm từ. Kích thước đầu vào là (None, 42).

- Lớp nhúng: Lớp này chuyển đổi mỗi token thành một vectơ số. Kích thước đầu ra của lớp nhúng là (None, 42, 200), trong đó 200 là kích thước của vectơ nhúng.
- Lớp LSTM: Lớp này là một mạng nơ-ron tái phát được sử dụng để học các mô hình dài hạn trong dữ liệu. Mô hình có ba lớp LSTM, mỗi lớp có 300 đơn vị ẩn. Kích thước đầu ra của lớp LSTM là [(None, 42, 300), (None, 300), (None, 300)], trong đó (None, 42, 300) là trạng thái ẩn của mô hình tại mỗi bước thời gian, (None, 300) là trạng thái ẩn cuối cùng của mô hình và (None, 300) là vectơ đầu ra của mô hình tại mỗi bước thời gian.
- Lớp thời gian phân bố (mật độ): Lớp này áp dụng một hàm mật độ điểm cho mỗi vectơ đầu ra của lớp LSTM. Kích thước đầu ra của lớp này là (None, None, 37523), trong đó None đại diện cho số lượng mẫu trong tập dữ liệu và 37523 đại diện cho số lượng lớp trong mô hình.

optimizer='rmsprop': Sử dụng thuật toán RMSprop (Root Mean Square Prop) để tối ưu hóa trọng số mô hình trong quá trình huấn luyện. RMSprop là một biến thể hiệu quả của Stochastic Gradient Descent (SGD) và điều chỉnh tốc độ học tập cho từng trọng số dựa trên gradient trước đó.

loss='sparse_categorical_crossentropy': Sử dụng hàm tổn thất "sparse_categorical_crossentropy" để đo lường độ lệch giữa giá trị dự đoán của mô hình và giá trị thực tế.

```
      model.compile(optimizer='rmsprop',loss='sparse_categorical_crossen
tropy')
```

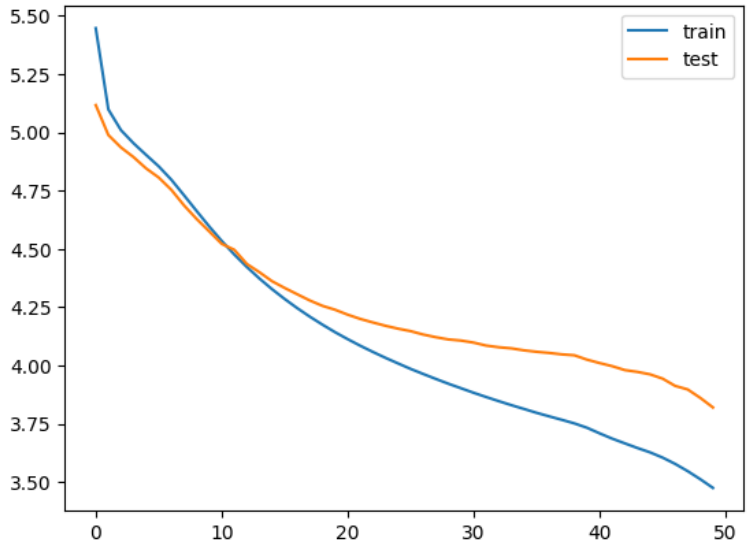
Tiến hành train model với 50 epoch.

```
history = model.fit(
          [x_tr, y_tr[:, :-1]],
          y_tr.reshape(y_tr.shape[0], y_tr.shape[1], 1)[:, 1:],
          epochs=50,
          callbacks=[es],
          batch_size=128,
          validation_data=([x_val, y_val[:, :-1]],
                           y_val.reshape(y_val.shape[0], y_val.shape[1],
1)[:
                           , 1:]),
          )
```

CHƯƠNG 4: KẾT QUẢ

4.1. Kết quả Train

Dưới đây là biểu diễn của hàm lỗi khi huấn luyện mô hình Sequence-to-Sequence. Bài đồ án sử dụng thư viện huấn luyện mô hình Sequence-to-Sequence with Attention cho bài toán tóm tắt văn bản tự động. Các thông số cần cân chỉnh là số lượng nơ-ron của RNN encoder và decoder phù hợp với độ dài của dữ liệu đầu vào. Số lượng câu lấy từ arcticle là max_arcticle_sentences=5, độ dài tối đa của đoạn abstract là max_abstract_sentences=100. Chúng tôi sử dụng Gradient Descent Optimizer với learning rate giảm từ 0,15 đến 0,000015 để tối thiểu hàm lỗi và Beam Search với độ rộng K=5 để sinh ra câu.



Hình 4.1. *Train và Test Loss*

Từ Hình 4.1 có nhận xét sau: Cả lỗi train và lỗi test đều có xu hướng giảm dần theo số epoch, cho thấy mô hình đang học tập hiệu quả và cải thiện khả năng dự đoán theo thời gian. Lỗi test luôn thấp hơn lỗi train, loại trừ khả năng quá khớp (overfitting) của mô hình.

Top các dự đoán có ROUGE-N cao nhất:

```
đoạn text: state bank india atm south delhis sangam vihar dispensed fake notes b
tóm tắt gốc: sbi atm dispenses fake notes churan lable strip
1/1 [==============================] - 0s 91ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 34ms/step
tóm tắt dự đoán: sbi atm dispenses churan lable notes fake notes
```

```
đoạn text: twinkle khanna slammed twitter user mocked husband akshay ku akshays meeting pr
tóm tắt gốc: twinkle slams twitter user mocking husband akshay
1/1 [==============================] - 0s 90ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 36ms/step
tóm tắt dự đoán: twinkle slams akshay calling twitter user mocking pm modi
```

```
đoạn text: mthembers sunrisers hyderabad squad wore lungis danced bollywood song lu
tóm tắt gốc: srh players wear lungis dance lungi dance
1/1 [==============================] - 0s 102ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 34ms/step
tóm tắt dự đoán: england players wear lungi dance dance lungi
```

```
đoạn text: sports minister vijay goel asked national antidoping agency consider
tóm tắt gốc: sports minister asks antidope body criminalise doping
1/1 [==============================] - 0s 95ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 36ms/step
tóm tắt dự đoán: sports ministry asks antidope body criminalise doping
```

4.2. Đánh giá kết quả

```
rouge1: 0.3109274891774892
rouge2: 0.1027406746031746
rougeL: 0.2871901875901876
```

Hình 4.2. *Kết quả ROUGE-1, ROUGE-2, ROUGE_L*

ROUGE-1= 0.3109 : ROUGE-1 đo lường tỷ lệ trùng khớp của các từ đơn lẻ giữa bản tóm tắt tự động và bản tóm tắt tham chiếu. Giá trị 0.3109 cho thấy rằng khoảng 31.09% từ trong bản tóm tắt tự động trùng với từ trong bản tóm tắt tham chiếu. Đây là một kết quả tương đối khả quan.

ROUGE-2 = 0.1027: ROUGE-2 đo lường tỷ lệ trùng khớp của các cặp từ (bigrams) giữa bản tóm tắt tự động và bản tóm tắt tham chiếu. Giá trị 0.1027 (khoảng 10.27%) cho thấy mức độ trùng khớp của các cặp từ là khá thấp.

ROUGE-L = 0.2872: ROUGE-L đo lường độ dài của chuỗi con chung dài nhất (longest common subsequence - LCS) giữa bản tóm tắt tự động và bản tóm tắt tham chiếu, từ đó đánh giá sự bảo toàn cấu trúc câu. Giá trị 0.2872 cho thấy rằng mô hình duy trì được khoảng 28.72% cấu trúc câu gốc.

Vậy các kết quả ROUGE-1, ROUGE-2, và ROUGE-L cho thấy rằng mô hình tóm tắt văn bản đã đạt được một số thành công trong việc nắm bắt các từ đơn lẻ và một phần cấu trúc câu, nhưng còn hạn chế trong việc tái hiện các cụm từ quan trọng. Để cải thiện ta cần tăng dữ liệu đa dạng các lĩnh vực vào tập train.

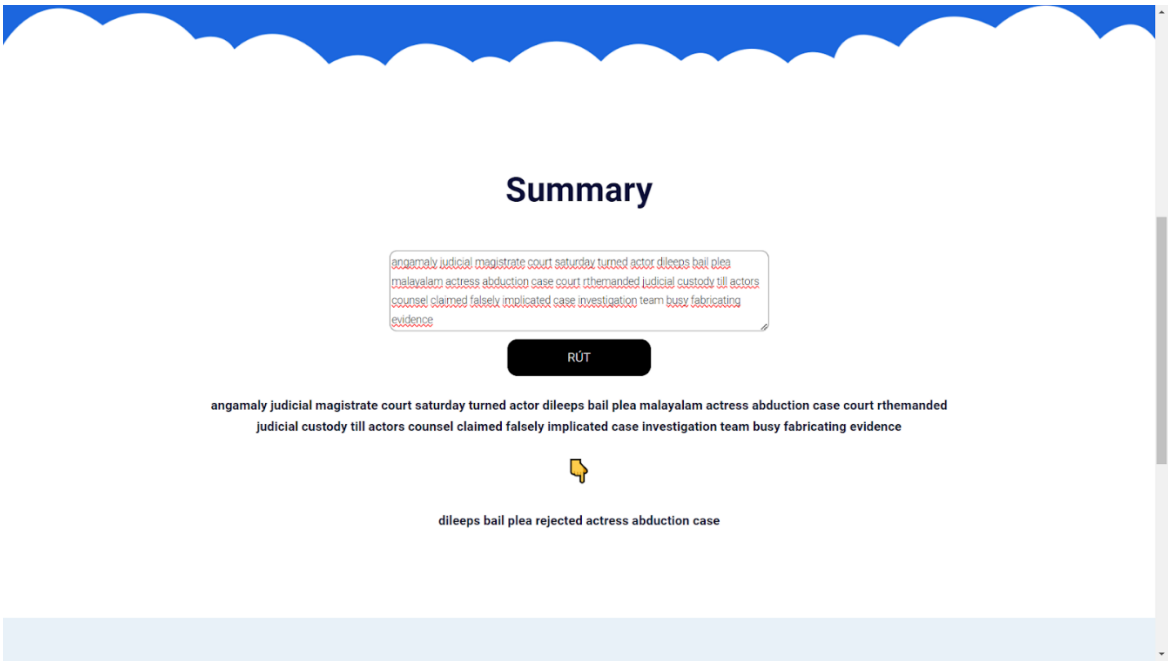
4.3. Triển khai Web App

Sử dụng thư viện Flask để xây dựng backend API cho ứng dụng. Về frontend dùng html css js và sử dụng github page để deploy chúng.

```
      ngrok.kill()
      ngrok.set_auth_token("1zxLGGOi15rdUwsjI1hY3r5jpQ6_383vhrdHUinWjsB
wyHms6")
      public_url = ngrok.connect("5000").public_url
      print("API link \n",public_url)
      app = Flask(__name__)
      cors = CORS(app)
      app.config['CORS_HEADERS'] = 'Content-Type'
      app.config['JSON_AS_ASCII'] = False
      run_with_ngrok(app)

      @app.route("/")
```

```
@cross_origin()
def test():
    return jsonify({"err":0})

@app.route('/summary', methods=['POST'] )
@cross_origin()
def summary():
    if not request.is_json:
        return jsonify({"error": "Request must be JSON"}), 400
    input_text = request.json.get('input_text')
    return jsonify({"summary": summaryAPI(input_text)})

app.run()
```

Giao diện tóm tắt văn bản và kết quả tóm tắt khi đưa vào một đoạn tiếng anh.



Hình 4.3. *Giao diện tóm tắt văn bản*

Từ Hình 4.3 thấy được, khi đưa một đoạn văn bản tiếng anh bất kì thì khi nhấn rút sẽ cho ra đoạn tóm tắt văn bản tương ứng.

KẾT LUẬN

Việc sử dụng mô hình Sequence-to-Sequence với Attention đã mở ra một hướng đi cho bài toán tóm tắt văn bản tự động. Đặc biệt, sự kết hợp mạnh mẽ giữa học sâu và xử lý ngôn ngữ tự nhiên đã được chứng minh là hiệu quả, góp phần đáng kể vào việc nâng cao độ chính xác của việc tóm tắt văn bản tiếng Anh. Các mô hình này không chỉ giúp máy tính hiểu rõ hơn về ngữ cảnh mà còn cải thiện khả năng chọn lọc thông tin quan trọng từ văn bản gốc để tạo ra các bản tóm tắt súc tích và chính xác. Tuy nhiên, để tiếp tục nâng cao hiệu quả và độ chính xác của mô hình, chúng ta cần chú trọng vào việc xây dựng một tập dữ liệu đầu vào Word2vec có độ chính xác cao hơn. Điều này đòi hỏi việc mô tả chi tiết và rõ ràng hơn mối quan hệ giữa các từ, giúp máy tính hiểu sâu hơn về ngữ nghĩa và cách sử dụng của từng từ trong các ngữ cảnh khác nhau. Việc này không chỉ đơn thuần là thu thập dữ liệu mà còn yêu cầu một quá trình xử lý và lọc dữ liệu tinh vi để đảm bảo chất lượng.

Chính vì thế, việc chuẩn bị một tập dữ liệu lớn và phong phú về mặt từ vựng trở nên vô cùng cần thiết cho việc phát triển một mô hình tóm tắt văn bản tự động tiếng Anh hiệu quả. Tập dữ liệu này cần bao gồm nhiều nguồn tài liệu khác nhau, từ các bài viết khoa học, báo chí, cho đến các tài liệu văn học và đối thoại hàng ngày. Điều này giúp mô hình có thể học hỏi từ nhiều ngữ cảnh và phong cách ngôn ngữ khác nhau, từ đó cải thiện khả năng tổng hợp và tóm tắt thông tin. Ngoài ra, việc liên tục cập nhật và mở rộng tập dữ liệu cũng rất quan trọng để theo kịp với sự thay đổi và phát triển của ngôn ngữ. Ngôn ngữ không ngừng biến đổi, và mô hình cũng cần phải thích nghi để có thể xử lý và tóm tắt các văn bản mới một cách chính xác và hiệu quả. Sự đa dạng và phong phú của tập dữ liệu đầu vào sẽ là nền tảng vững chắc giúp mô hình tóm tắt văn bản tự động tiếng Anh trở nên ngày càng hoàn thiện và mạnh mẽ hơn.

TÀI LIỆU THAM KHẢO

[1] "Accern • Articles & Resources • What is NLP Text Summarization: Benefits & Use Cases." Accessed: Jun. 05, 2024. [Online]. Available: https://www.accern.com/resources/what-is-nlp-text-summarization-benefits-use-cases

[2] "Seq2Seq Model | Understand Seq2Seq Model Architecture." Accessed: Jun. 05, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/

[3] "Introduction to Recurrent Neural Network - GeeksforGeeks." Accessed: Jun. 05, 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[4] S. Santhosh, "Understanding BLEU and ROUGE score for NLP evaluation," Medium. Accessed: Jun. 05, 2024. [Online]. Available: https://medium.com/@sthanikamsanthosh1994/understanding-bleu-and-rouge-score-for-nlp-evaluation-1ab334ecadcb

[5] B. biên tập TopDev, "Flask python là gì? - Những điều cần biết," TopDev. Accessed: Jun. 05, 2024. [Online]. Available: https://topdev.vn/blog/flask-python-la-gi-nhung-dieu-can-biet/