

TRƯỜNG ĐẠI HỌC THỦY LỢI



ĐỀ TÀI:

PHÂN LOẠI BÀI VIẾT TIN TỨC BẰNG LSTM

Sinh viên thực hiện: Đinh Quốc Việt, 64TTNT.NB

Khoa: Công nghệ thông tin

Giáo viên hướng dẫn: PGS.TS. Nguyễn Quang Hoan

Hà Nội, Tháng 6 năm 2025

Mục Lục

I.MỞ ĐẦU.....	3
I.1.Tổng quan tình hình nghiên cứu thuộc lĩnh vực đề tài.....	3
I.2.Lý do chọn đề tài.....	3
I.3.Mục tiêu đề tài.....	4
I.4.Phương pháp nghiên cứu.....	4
I.5.Đối tượng và phạm vi nghiên cứu.....	4
I.5.1.Đối tượng.....	4
I.5.2.Phạm vi nghiên cứu.....	4
II. KHÁM PHÁ DỮ LIỆU.....	5
II.1 Tổng quan dữ liệu.....	5
II.2 Trực quan hóa dữ liệu.....	6
III. TIỀN XỬ LÝ DỮ LIỆU.....	9
III.1 Tổng hợp văn bản đầu vào và làm sạch văn bản.....	9
III.2 Tách từ và tạo từ điển.....	11
III.3 Padding chuỗi.....	12
III.4 Xử lý nhãn đầu ra.....	13
V. LLM VÀ RAG CHO ỨNG DỤNG CHATBOT.....	15
V.1: LLM hỗ trợ cho ứng dụng chatbot.....	15
VI. XÂY DỰNG ỨNG DỤNG.....	16
VI.1. Nội dung nghiên cứu.....	16
VI.2. Kết quả nghiên cứu đạt được.....	16
VII.KẾT LUẬN VÀ KIẾN NGHỊ.....	20
VII.1.Kết Luận.....	20
VII.2.Kiến nghị.....	21
VIII.TÀI LIỆU THAM KHẢO.....	23
VIII.1.Danh mục link tham khảo.....	23
VIII.2.Danh mục sách tham khảo.....	23

I.MỞ ĐẦU

I.1.Tổng quan tình hình nghiên cứu thuộc lĩnh vực đề tài

Trong thời đại bùng nổ thông tin như hiện nay, mỗi ngày có hàng triệu bài viết tin tức được xuất bản trên các nền tảng báo chí điện tử, mạng xã hội và blog. Việc tự động phân loại các bài viết này thành các chủ đề như chính trị, thể thao, kinh tế, giải trí, khoa học... đóng vai trò quan trọng trong việc quản lý, tìm kiếm và cá nhân hóa nội dung cho người dùng.

Trước đây, việc phân loại văn bản thường dựa vào các phương pháp học máy truyền thống như Naive Bayes, SVM hay Decision Tree. Tuy nhiên, các phương pháp này gặp hạn chế khi xử lý các đặc điểm ngữ cảnh phức tạp và thứ tự từ trong câu. Với sự phát triển mạnh mẽ của các mô hình học sâu, đặc biệt là mạng Long Short-Term Memory (LSTM) — một biến thể của mạng nơ-ron hồi tiếp (RNN), khả năng xử lý ngôn ngữ tự nhiên đã được cải thiện đáng kể, đặc biệt trong các bài toán tuần tự như phân tích cảm xúc, dịch máy và phân loại văn bản.

Hiện nay, việc ứng dụng LSTM vào bài toán phân loại văn bản, đặc biệt là phân loại tin tức, đã và đang cho thấy hiệu quả cao nhờ khả năng ghi nhớ thông tin dài hạn và khai thác mối quan hệ ngữ nghĩa giữa các từ trong câu. Do đó, việc nghiên cứu và triển khai mô hình LSTM để phân loại tin tức là hướng đi phù hợp và có tính thực tiễn cao.

I.2.Lý do chọn đề tài

Với khối lượng tin tức khổng lồ mỗi ngày, người dùng rất khó tiếp cận thông tin đúng với mối quan tâm của mình nếu không có hệ thống phân loại hiệu quả. Các công cụ phân loại hiện tại chưa tận dụng triệt để sức mạnh của các mô hình học sâu, hoặc chưa được tối ưu cho tiếng Việt. Vì vậy tôi chọn đề tài này với mong muốn: Áp dụng công nghệ trí tuệ nhân tạo hiện đại để giải quyết bài toán thực tế. Nâng cao trải

nghiệm người dùng trong việc tiếp cận tin tức. Mở rộng hiểu biết về ứng dụng LSTM trong xử lý ngôn ngữ tự nhiên.

I.3.Mục tiêu đề tài

Nghiên cứu mô hình LSTM và cách áp dụng vào bài toán phân loại văn bản. Thu thập, tiền xử lý và xây dựng tập dữ liệu tin tức theo các danh mục chủ đề cụ thể. Huấn luyện mô hình phân loại tin tức dựa trên dữ liệu tiếng Việt. Đánh giá hiệu quả của mô hình thông qua các chỉ số như accuracy, precision, recall và F1-score. Triển khai mô hình thành một hệ thống phân loại tin tức đơn giản.

I.4.Phương pháp nghiên cứu

Tổng quan lý thuyết: Nghiên cứu các tài liệu, bài báo khoa học liên quan đến LSTM và phân loại văn bản. Tiền xử lý dữ liệu: Làm sạch văn bản, chuẩn hóa, tách từ (tokenization), mã hóa bằng các kỹ thuật như one-hot, embedding. Xây dựng mô hình: Sử dụng thư viện TensorFlow/Keras để xây dựng mô hình LSTM. Huấn luyện và đánh giá: Chia dữ liệu thành tập huấn luyện và kiểm thử để đánh giá hiệu quả mô hình. Triển khai: Thiết kế giao diện đơn giản cho phép nhập văn bản và hiển thị kết quả phân loại.

I.5.Đối tượng và phạm vi nghiên cứu

I.5.1.Đối tượng

Các bài viết tin tức dạng văn bản, chủ yếu ở định dạng văn bản ngắn đến trung bình (từ 100–500 từ). Các mô hình học sâu, đặc biệt là LSTM, dùng trong xử lý ngôn ngữ tự nhiên.

I.5.2.Phạm vi nghiên cứu

Phân loại các bài viết tin tức thành một số chủ đề cụ thể như: thời sự, thể thao, giải trí, kinh tế... Chỉ tập trung vào nội dung văn bản, không xử lý hình ảnh hay metadata đi kèm. Mô hình LSTM huấn luyện từ đầu.

II. KHÁM PHÁ DỮ LIỆU

II.1 Tổng quan dữ liệu

Bộ dữ liệu AG News Classification Dataset là một tập hợp các bài báo tiếng Anh thu thập từ các nguồn tin tức lớn, được phát triển để phục vụ các bài toán phân loại văn bản. Đây là một phiên bản con của tập AG's News Corpus

Nguồn dữ liệu được lấy từ:

<https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset>

	Class Index	Title	Description
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil exportf...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...

Dữ liệu bao gồm ba cột chính:

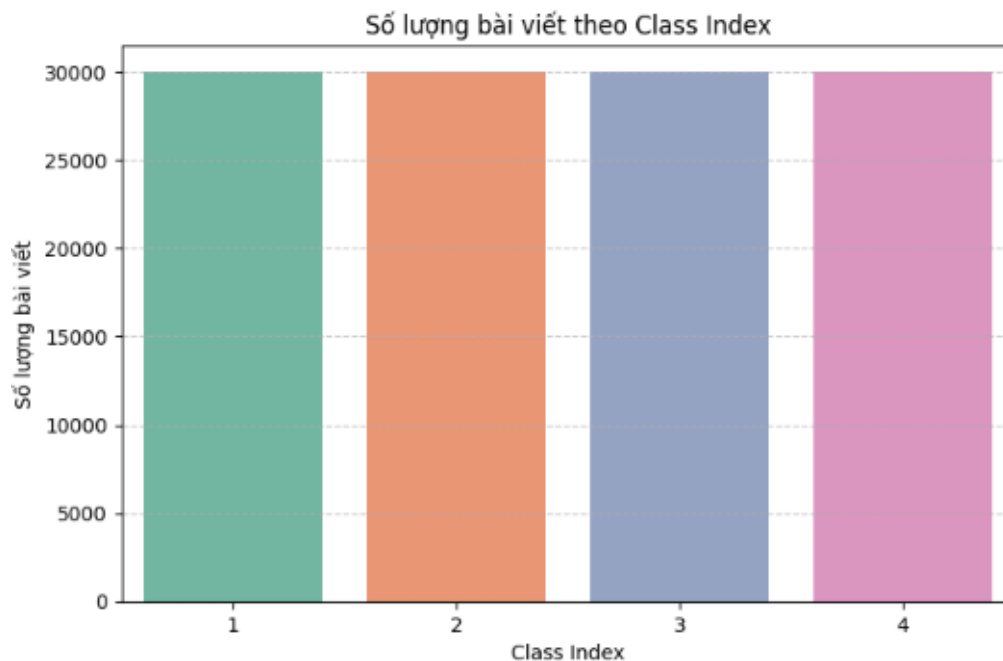
- Class Index: Cột thể hiện nhãn phân loại của bài viết, giá trị là số nguyên từ 1 đến 4 tương ứng với chủ đề Business (Kinh doanh) trong bộ dữ liệu AG News.
- Title: Tiêu đề ngắn gọn của bài viết, mang tính tóm tắt nội dung chính.
- Description: Mô tả chi tiết hơn về nội dung bài báo, thường là đoạn trích hoặc nội dung mở đầu.

```
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Class Index  120000 non-null int64
1   Title        120000 non-null object
2   Description  120000 non-null object
```

STT	Tên cột	Kiểu dữ liệu	Mô tả
0	Class Index	int64	Nhãn phân loại của bài viết, gồm 4 lớp chính: 1 = World, 2 = Sports, 3 = Business, 4 = Science/Technology
1	Title	object (string)	Tiêu đề ngắn gọn của bài báo. Dùng để tóm tắt nội dung chính.
2	Description	object (string)	Mô tả chi tiết hoặc phần mở đầu của bài báo. Thường được dùng làm đầu vào chính cho mô hình phân loại.

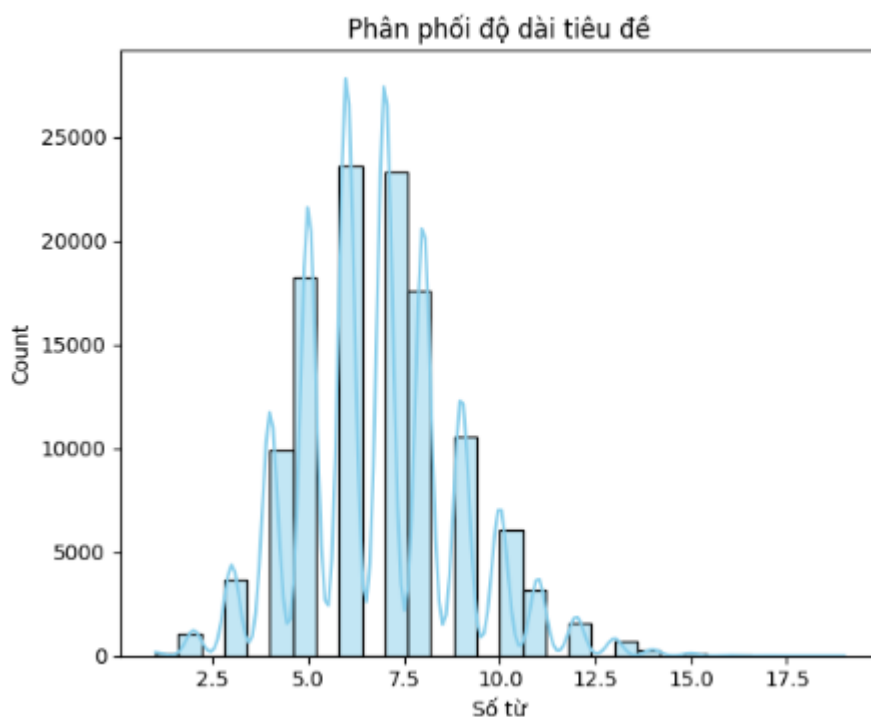
II.2 Trực quan hóa dữ liệu

- Số lượng bài viết theo Class Index



Mỗi Class Index tương ứng với một chủ đề tin tức: 1: World (Thế giới), 2: Sports (Thể thao), 3: Business (Kinh doanh), 4: Science/Technology (Khoa học/Công nghệ). Dữ liệu đã được cân bằng tốt: Mỗi nhãn có số lượng bài viết xấp xỉ 30.000, thể hiện qua chiều cao các cột gần như bằng nhau. Phân phối đồng đều: Không có nhãn nào chiếm ưu thế hoặc thiếu hụt đáng kể. Đây là đặc điểm lý tưởng cho các mô hình phân loại, giúp tránh hiện tượng mất cân bằng lớp (class imbalance) — vốn có thể làm mô hình thiên lệch và giảm độ chính xác. Phù hợp cho supervised learning: Việc có tập huấn luyện cân bằng giúp mô hình học được đặc trưng của từng lớp tốt hơn, đặc biệt với các kiến trúc như LSTM trong bài toán phân loại văn bản.

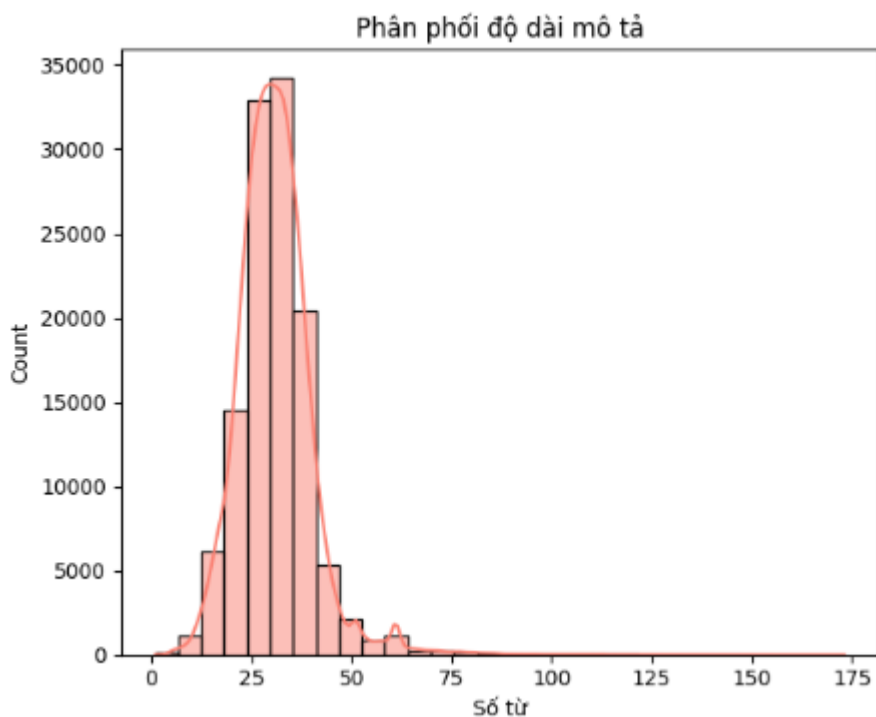
- Phân phối độ dài tiêu đề:



Trục hoành biểu thị số từ trong tiêu đề (Title). Phần lớn tiêu đề có độ dài từ 5 đến 9 từ. Hình dạng phân phối gần giống chuẩn, tuy nhiên có các đỉnh lặp thể hiện dữ

liệu dạng rời rạc (vì số từ là số nguyên). Có một số ít tiêu đề rất ngắn (1–3 từ) hoặc dài trên 12 từ, nhưng số lượng không đáng kể.

- Phân phối độ dài mô tả:



Trục hoành thể hiện số từ trong mô tả bài viết (Description). Mô tả có xu hướng dài hơn, phần lớn nằm trong khoảng 25 đến 45 từ, đỉnh khoảng 35 từ. Có đuôi kéo dài về bên phải (phân phối lệch phải), với một số mô tả lên tới 100+ từ, thậm chí hơn 150 từ (ít gặp).

III. TIỀN XỬ LÝ DỮ LIỆU

III.1 Tổng hợp văn bản đầu vào và làm sạch văn bản

Để tăng lượng thông tin ngữ nghĩa và cải thiện hiệu quả phân loại, hai trường văn bản là Title (tiêu đề) và Description (mô tả) được kết hợp lại thành một câu văn bản duy nhất. Việc này giúp mô hình tận dụng được cả thông tin tóm tắt và chi tiết cho mỗi bài báo.

	text
0	Wall St. Bears Claw Back Into the Black (Reute...
1	Carlyle Looks Toward Commercial Aerospace (Reu...
2	Oil and Economy Cloud Stocks' Outlook (Reuters...
3	Iraq Halts Oil Exports from Main Southern Pipe...
4	Oil prices soar to all-time record, posing new...
...	...

Việc tổng hợp này là bước đầu tiên trong chuỗi xử lý văn bản, trước khi làm sạch, tokenization và đưa vào mô hình học máy.

Sau khi tổng hợp nội dung từ hai trường Title và Description, bước tiếp theo là làm sạch văn bản nhằm loại bỏ các yếu tố gây nhiễu không cần thiết và chuẩn hóa dữ liệu đầu vào trước khi đưa vào mô hình học máy. Cụ thể, các thao tác làm sạch được thực hiện bao gồm:

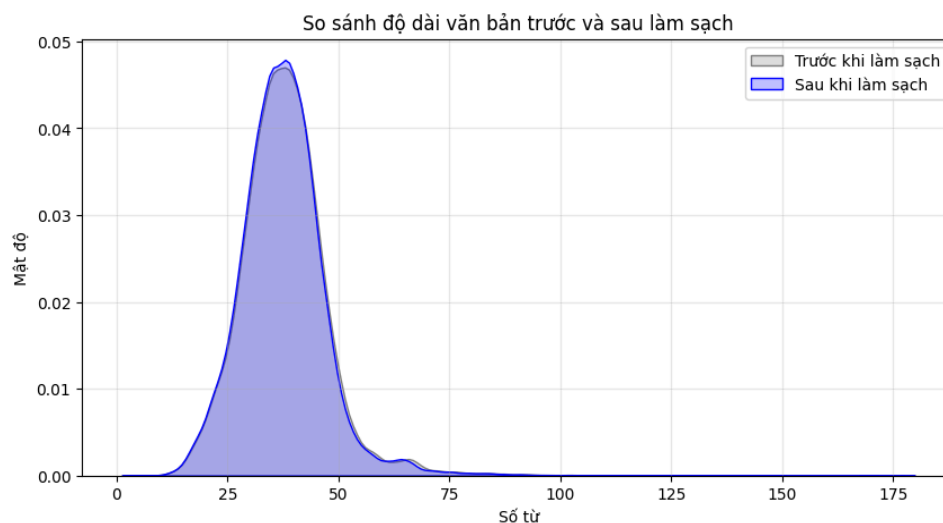
STT	Bước xử lý	Mô tả chi tiết
1	Chuyển chữ thường	Chuyển toàn bộ văn bản sang chữ thường để thống nhất định dạng văn bản.
2	Loại bỏ ký tự escape (\xa0, \n)	Xóa các ký tự không hiển thị thường gặp trong văn bản thu thập từ web hoặc file.

3	Xóa URL	Loại bỏ các đường dẫn (link) gây nhiễu trong câu, thường bắt đầu bằng http://.
4	Loại bỏ ký tự đặc biệt & dấu câu	Xóa các ký hiệu như !@#\$%^&*()[]{};,:.<>? để giữ lại từ ngữ có giá trị ngữ nghĩa.
5	Chuẩn hóa khoảng trắng	Rút gọn các khoảng trắng dư thừa về một khoảng trắng duy nhất.

Sau khi làm sạch dữ liệu:

	text	clean_text
71787	BBC set for major shake-up, claims newspaper L...	bbc set for major shakeup claims newspaper lon...
67218	Marsh averts cash crunch Embattled insurance b...	marsh averts cash crunch embattled insurance b...
54066	Jeter, Yankees Look to Take Control (AP) AP - ...	jeter yankees look to take control ap ap derek...
7168	Flying the Sun to Safety When the Genesis caps...	flying the sun to safety when the genesis caps...
29618	Stocks Seen Flat as Nortel and Oil Weigh NEW ...	stocks seen flat as nortel and oil weigh new y...

Trong cột clean_text, toàn bộ văn bản đã được chuyển về chữ thường, giúp thống nhất định dạng.



Biểu đồ trên thể hiện phân phối số từ trong mỗi văn bản trước và sau khi làm sạch, với trục hoành là số từ và trục tung là mật độ phân phối. Sau khi chúng ta làm sạch dữ liệu ta thấy được một số văn bản ở khoảng 35 đã giảm tức là đã có bước làm sạch loại bỏ các ký tự dư thừa đi.

III.2 Tách từ và tạo từ điển

Sau khi làm sạch văn bản, bước tiếp theo là Tokenization(Bộ tách từ) – quá trình chia văn bản thành các đơn vị nhỏ hơn gọi là “token”, thường là từ hoặc cụm từ. Đây là bước quan trọng nhằm chuẩn bị dữ liệu đầu vào cho mô hình học máy.

Tokenizer có nhiệm vụ: Tách văn bản thành danh sách các từ riêng biệt (tokens). Loại bỏ các yếu tố không cần thiết (như khoảng trắng thừa). Giữ nguyên thứ tự từ – điều rất quan trọng trong các mô hình như LSTM. Ví dụ: Văn bản gốc: "Stocks rise amid positive earnings" → Tokenized: ["stocks", "rise", "amid", "positive", "earnings"]

Sau khi đã có danh sách token, bước tiếp theo là xây dựng từ điển ánh xạ từ → số nguyên. Điều này là cần thiết vì các mô hình học sâu như LSTM chỉ có thể xử lý dữ liệu số, không hiểu được ký tự chữ cái. Ví dụ:

Từ (token)	Số đại diện
stocks	1
rise	2
amid	3
positive	4

Với ánh xạ này, câu đầu vào sẽ trở thành: arduino Sao chép Chính sửa "Stocks rise amid positive earnings" → [1, 2, 3, 4, 5]

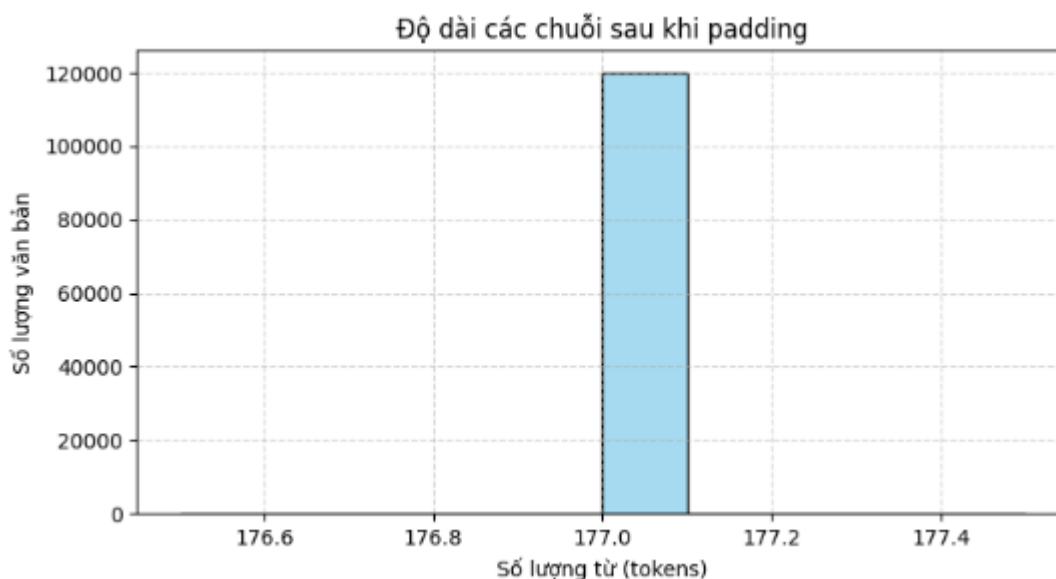
Sau khi tách từ và tạo từ điển kết quả ta thu được số lượng từ vựng: 96187. Điều này có nghĩa là trong toàn bộ tập dữ liệu có 96,187 từ duy nhất (không trùng lặp) sau khi làm sạch và tách từ.

III.3 Padding chuỗi

Trong xử lý ngôn ngữ tự nhiên (NLP), sau khi văn bản được chuyển thành chuỗi số nguyên thông qua bước tokenization, mỗi câu lại có độ dài khác nhau tùy theo nội dung. Tuy nhiên, mô hình học sâu như LSTM yêu cầu đầu vào phải có kích thước đồng nhất, tức là tất cả chuỗi phải có cùng số lượng phần tử. Do đó, ta cần thực hiện padding – tức là đệm thêm các giá trị 0 vào đầu hoặc cuối của chuỗi để chuẩn hóa độ dài. Ví dụ:

Văn bản gốc	Sau Tokenizer	Sau Padding (maxlen=6)
"stock rise now"	[12, 34, 56]	[12, 34, 56, 0, 0, 0]
"inflation slows economy"	[78, 21, 90, 65]	[78, 21, 90, 65, 0, 0]
"interest rate hike expected"	[9, 4, 99, 45, 18]	[9, 4, 99, 45, 18, 0]

Mục đích của padding: Đảm bảo đầu vào có cùng chiều để xử lý trong mạng neural. Giúp huấn luyện theo batch hiệu quả hơn, thay vì xử lý từng câu riêng lẻ. Hạn chế lỗi kích thước khi đưa vào các mô hình học sâu như LSTM, GRU, Transformer.



Tất cả các văn bản đều có độ dài đúng bằng 177 tokens. Điều này chứng tỏ padding đã được thực hiện thành công và chuẩn hóa độ dài chuỗi một cách nhất quán. Việc chuẩn hóa này rất cần thiết để mô hình học sâu như LSTM hoặc GRU có thể xử lý dữ liệu đầu vào theo batch (lô) hiệu quả.

III.4 Xử lý nhãn đầu ra

Trong bài toán phân loại văn bản, đầu ra của mô hình là một nhãn phân loại (label), ví dụ như “Thể giới”, “Thể thao”, “Kinh doanh”, “Khoa học/Công nghệ”. Trong bộ dữ liệu AG News, các nhãn này ban đầu được biểu diễn bằng chỉ số nguyên trong cột Class Index, với các giá trị từ 1 đến 4. Vì các mô hình học máy thường sử dụng nhãn bắt đầu từ 0, nên ta tiến hành chuẩn hóa như sau: Chuẩn hóa nhãn về 0–3

Class Index (gốc)	Nhãn sau chuẩn hóa
1	0
2	1

3	2
4	3

Để phù hợp với đầu ra của mô hình LSTM sử dụng hàm kích hoạt softmax, ta cần chuyển các nhãn số thành vector nhị phân gọi là one-hot vector. Chúng ta sẽ sử dụng One-hot Encoding

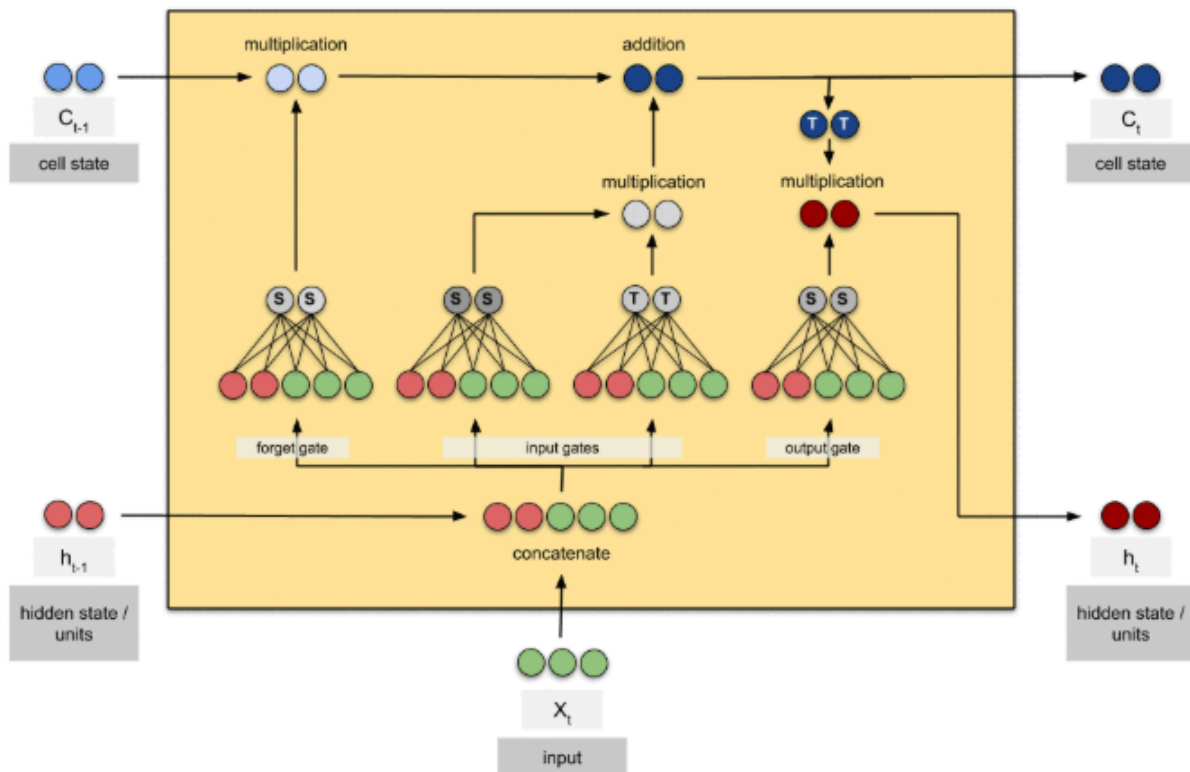
Nhãn số	One-hot vector
0	[1, 0, 0, 0]
1	[0, 1, 0, 0]
2	[0, 0, 1, 0]
3	[0, 0, 0, 1]

Chuẩn hóa nhãn giúp mô hình học tốt hơn và tránh lỗi chỉ số. One-hot encoding giúp mô hình nhận diện chính xác lớp nào được dự đoán. Đây là bước cầu nối giữa dữ liệu thô và mô hình học sâu.

IV. XÂY DỰNG MÔ HÌNH LSTM CHO PHÂN LOẠI BÀI VIẾT TIN TỨC

IV.1 Tổng quan về LSTM

LSTM được thiết kế để vượt qua nhược điểm của RNN truyền thống là quên thông tin trong các chuỗi dài. Nó làm điều này thông qua ba cổng chính: Forget gate – Cổng quên, Input gate – Cổng đầu vào, Output gate – Cổng đầu ra.



Các bước của hình trên sẽ như sau:

Đầu vào x_t : vector đầu vào ở thời điểm hiện tại, h_{t-1} : trạng thái ẩn từ bước trước, C_{t-1} : trạng thái cell từ bước trước. Cả x_t và h_{t-1} được nối lại (concatenate) để làm đầu vào cho các cổng.

Forget Gate (Cổng quên) Mục tiêu: quyết định thông tin nào trong C_{t-1} sẽ bị quên. Biểu thức: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$. Đầu ra f_t là một vector có giá trị từ 0 đến 1, nhân với C_{t-1} .

Input Gate(Cổng đầu vào) Gồm hai phần: i_t : quyết định phần nào của thông tin mới sẽ được ghi nhớ. $C \sim t$: thông tin ứng viên được tạo từ h_{t-1} , x_t .

Cập nhật trạng thái cell: Kết hợp thông tin quên và thông tin mới: $C_t = f_t * C_{t-1} + i_t * C \sim t$

Output Gate (Cổng đầu ra): Quyết định trạng thái ẩn mới h_t : $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \Rightarrow h_t = o_t * \tanh(C_t)$

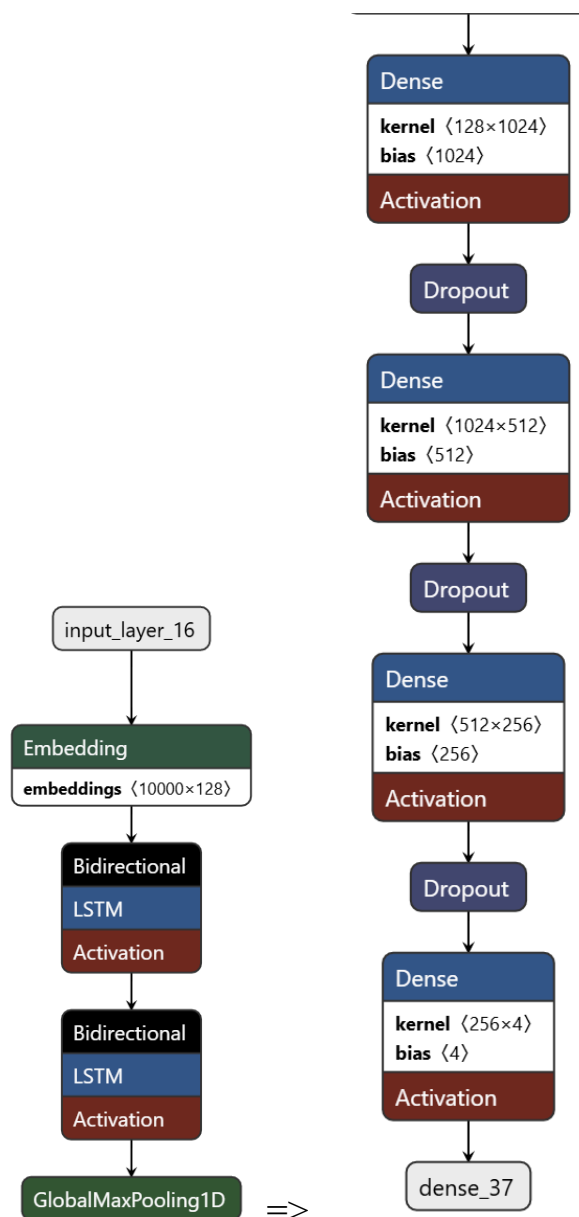
Đầu ra: h_t : trạng thái ẩn (dùng cho bước tiếp theo hoặc output) và C_t : trạng thái cell (được chuyển qua bước kế tiếp)

IV.2 Xây dựng mô hình LSTM

Khi xây dựng một mô hình học sâu như LSTM để phân loại văn bản, việc lựa chọn và điều chỉnh các siêu tham số (hyperparameters) là cực kỳ quan trọng. Dưới đây là các tham số chính trong mô hình đã xây dựng:

Tên tham số	Giá trị	Mô tả
VOCAB_SIZE	10,000	Số từ phổ biến nhất dùng trong từ điển; từ hiếm bị gán thành <OOV>
MAXLEN	177	Độ dài tối đa của mỗi chuỗi văn bản sau padding/truncating
embed_size	128	Kích thước vector embedding cho mỗi từ
LSTM units	128 \rightarrow 64 (Bidirectional)	Hai lớp LSTM hai chiều, lần lượt có 128 và 64 đơn vị

Pooling	GlobalMaxPooling1D	Lấy giá trị cực đại trên toàn bộ chuỗi thời gian
Dense layers	1024 \rightarrow 512 \rightarrow 256	Các lớp fully-connected, giảm dần số đơn vị
Dropout	0.25 sau mỗi Dense	Tắt ngẫu nhiên 25% số neurons để tránh overfitting
Output layer	4 lớp (softmax)	Phân loại văn bản thành 4 lớp khác nhau
batch_size	256	Số mẫu xử lý trong mỗi bước huấn luyện
epochs	20	Số vòng lặp toàn bộ dữ liệu huấn luyện
EarlyStopping	patience = 4	Dừng sớm nếu val_loss không giảm sau 4 epoch
optimizer	Adam (lr=1e-4)	Tối ưu hóa hàm mất mát với tốc độ học thấp (ổn định hơn)
loss function	categorical_crossentropy	Dùng cho phân loại đa lớp (multi-class classification)



Mô hình bắt đầu với lớp đầu vào, nơi chuỗi văn bản được chuyển đổi thành các chỉ số số nguyên, đại diện cho từng từ hoặc token trong câu. Những chỉ số này sau đó được đưa vào lớp Embedding, nơi mỗi từ được ánh xạ thành một vector có kích thước 128 chiều. Lớp Embedding này giúp mô hình hiểu được ngữ nghĩa và mối quan hệ giữa các từ thông qua không gian vector liên tục.

Tiếp theo, chuỗi các vector được đưa vào hai lớp LSTM hai chiều (Bidirectional LSTM) liên tiếp. Việc sử dụng mạng LSTM hai chiều cho phép mô hình học được

thông tin ngữ cảnh không chỉ từ trái sang phải mà còn từ phải sang trái, nhờ đó hiểu rõ hơn ý nghĩa tổng thể của câu. Mỗi lớp LSTM được kết hợp với một hàm kích hoạt nhằm tăng tính phi tuyến cho mạng.

Sau khi đi qua hai lớp LSTM, đầu ra là một chuỗi các vector theo thời gian. Để tóm tắt toàn bộ chuỗi thành một vector duy nhất có ý nghĩa, mô hình sử dụng GlobalMaxPooling1D — lớp này chọn giá trị lớn nhất ở mỗi chiều trong toàn bộ chuỗi thời gian, từ đó tạo ra một vector biểu diễn tốt nhất cho cả câu.

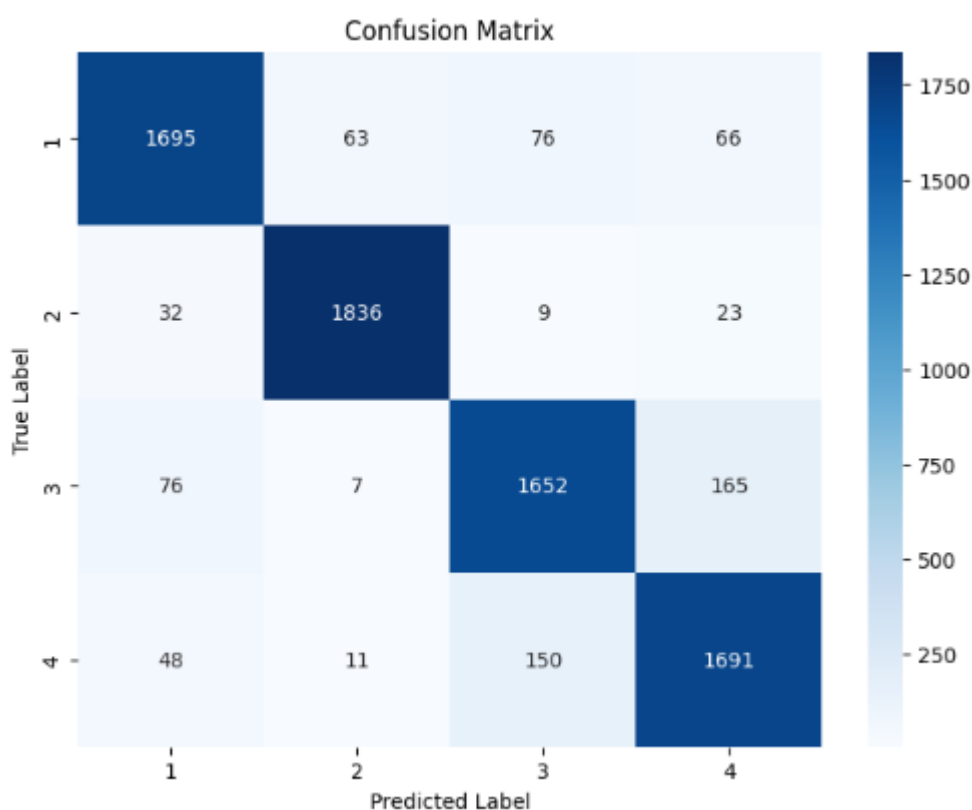
Từ đây, vector đặc trưng thu được được đưa qua một chuỗi các lớp Dense (fully connected) nhằm trích xuất thông tin sâu hơn và học các biểu diễn trừu tượng. Đầu tiên là lớp Dense có 1024 đơn vị, tiếp theo là 512 đơn vị, rồi đến 256 đơn vị, và cuối cùng là lớp đầu ra với 4 đơn vị tương ứng với số lượng lớp phân loại. Giữa các lớp Dense là các lớp Dropout, được sử dụng nhằm giảm hiện tượng quá khớp (overfitting) bằng cách ngẫu nhiên vô hiệu hóa một số đơn vị trong quá trình huấn luyện. Cuối cùng, đầu ra của lớp Dense cuối cùng đi qua một hàm kích hoạt (thường là softmax) để tạo ra xác suất cho từng lớp đầu ra, cho phép mô hình đưa ra dự đoán lớp phù hợp nhất với văn bản đầu vào.

IV.3 Kết quả mô hình LSTM

```
Epoch 1/20
469/469 ————— 44s 86ms/step - accuracy: 0.5000 - loss: 1.0563 - val_accuracy: 0.8491 - val_loss: 0.4482
Epoch 2/20
469/469 ————— 39s 83ms/step - accuracy: 0.8819 - loss: 0.3630 - val_accuracy: 0.8930 - val_loss: 0.3285
Epoch 3/20
469/469 ————— 41s 84ms/step - accuracy: 0.9129 - loss: 0.2725 - val_accuracy: 0.8996 - val_loss: 0.3045
Epoch 4/20
469/469 ————— 41s 84ms/step - accuracy: 0.9227 - loss: 0.2424 - val_accuracy: 0.9003 - val_loss: 0.3007
Epoch 5/20
469/469 ————— 41s 83ms/step - accuracy: 0.9334 - loss: 0.2128 - val_accuracy: 0.9021 - val_loss: 0.2974
Epoch 6/20
469/469 ————— 41s 84ms/step - accuracy: 0.9377 - loss: 0.1990 - val_accuracy: 0.9022 - val_loss: 0.2974
Epoch 7/20
469/469 ————— 40s 83ms/step - accuracy: 0.9437 - loss: 0.1793 - val_accuracy: 0.9045 - val_loss: 0.2869
Epoch 8/20
469/469 ————— 39s 83ms/step - accuracy: 0.9497 - loss: 0.1621 - val_accuracy: 0.9039 - val_loss: 0.2998
Epoch 9/20
469/469 ————— 41s 83ms/step - accuracy: 0.9545 - loss: 0.1482 - val_accuracy: 0.9033 - val_loss: 0.3054
Epoch 10/20
469/469 ————— 41s 83ms/step - accuracy: 0.9599 - loss: 0.1328 - val_accuracy: 0.9034 - val_loss: 0.3102
Epoch 11/20
469/469 ————— 39s 83ms/step - accuracy: 0.9622 - loss: 0.1245 - val_accuracy: 0.9058 - val_loss: 0.3214
Epoch 11: early stopping
Restoring model weights from the end of the best epoch: 7.
238/238 ————— 3s 12ms/step - accuracy: 0.9001 - loss: 0.2955
Test Accuracy: 0.9045
```

Sau epoch 7, mặc dù độ chính xác huấn luyện vẫn tiếp tục tăng, độ chính xác trên tập kiểm tra có dấu hiệu chững lại và không cải thiện thêm. Thậm chí, giá trị hàm mất mát trên tập validation bắt đầu tăng nhẹ, cho thấy mô hình bắt đầu ghi nhớ quá nhiều chi tiết của dữ liệu huấn luyện. Nhờ vào kỹ thuật dừng sớm (early stopping), quá trình huấn luyện đã được kết thúc kịp thời tại epoch 11 và mô hình được khôi phục về trạng thái tốt nhất ở epoch 7. Cuối cùng, khi đánh giá trên tập kiểm tra độc lập, mô hình đạt độ chính xác khoảng 90%, cho thấy khả năng tổng quát hóa tốt và hiệu quả phân loại cao. Nhìn chung, đây là một mô hình huấn luyện thành công với tốc độ hội tụ nhanh, hiệu suất cao và kiểm soát tốt hiện tượng overfitting.

Ma trận nhầm lẫn của mô hình trên tập test:



Nhìn vào ma trận, có thể thấy rằng mô hình hoạt động khá hiệu quả với tỷ lệ dự đoán đúng cao trên tất cả các lớp. Đặc biệt, lớp thứ hai đạt độ chính xác rất cao, khi

hầu như toàn bộ mẫu đều được phân loại đúng, cho thấy mô hình nhận diện lớp này tốt và ổn định.

Tổng thể, mô hình cho thấy khả năng phân loại tốt, đặc biệt là với các lớp có đặc trưng nổi bật và dễ phân biệt. Tuy nhiên, để cải thiện hơn nữa, đặc biệt là với các lớp dễ gây nhầm lẫn, có thể cân nhắc bổ sung thêm dữ liệu huấn luyện, tinh chỉnh kiến trúc mô hình, hoặc áp dụng các kỹ thuật như attention hoặc loss chuyên biệt để tăng cường khả năng phân biệt giữa các lớp gần nhau.

VI. XÂY DỰNG ỨNG DỤNG

Phần này sẽ là phần để trình bày một trang web mà người dùng có thể nhập vào Title và Description để xem nó thuộc loại nào trong 4 loại mà mô hình đã học.

Dự đoán thể loại bài báo



Tiêu đề (Title):

Scientists Discover Room-Temperature Superconductor

Mô tả (Description):

A team at Harvard University claims to have created the first superconductor that works at room temperature and pressure, a breakthrough that could revolutionize the energy and electronics industries, opening up the potential for lossless power transmission on a large scale.

Dự đoán thể loại

 **Kết quả dự đoán:** Class 4
 **Độ tin cậy:** 0.9645

Như kết quả đã cho chúng ta đã nhập vào một thông tin liên quan đến khoa học và mô hình đã trả ra đúng kết quả chúng ta mong đợi với độ tin cậy rất cao.

Dự đoán thể loại bài báo


Tiêu đề (Title):


Apple Reports Record First-Quarter Profit

Mô tả (Description):

Tech giant Apple has just released its financial report showing revenue of \$120 billion in the first quarter of 2025, exceeding all analysts' expectations. Profits increased sharply thanks to iPhone sales and cloud services. Apple's stock price jumped 6% immediately after the news.

Dự đoán thể loại

 **Kết quả dự đoán:** Class 4

 **Độ tin cậy:** 0.535

Lần này thử nghiệm khác với dạng dữ liệu về kinh doanh nhưng mô hình lại cho ra kết quả là thể thao. Có thể thấy vốn từ vựng mà mô hình học được trong bộ huấn luyện chưa đủ nhiều để có thể dự đoán được các loại phức tạp hơn

VII.KẾT LUẬN VÀ KIẾN NGHỊ

VII.1.Kết Luận

VII.2.Kiến nghị

VIII.TÀI LIỆU THAM KHẢO

- [1] Tập dữ liệu PAD-UFES-20: <https://data.mendeley.com/datasets/zr7vgbcyr2/1>