

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
TP.HỒ CHÍ MINH



HCMUTE

BÁO CÁO TRÍ TUỆ NHÂN TẠO

MÃ MÔN HỌC: ARIN337629

ĐỀ TÀI: PHÂN LOẠI VĂN BẢN TIẾNG VIỆT

Sinh viên thực hiện : VÕ HOÀNG AN
Lớp : 19146CL2A
MSSV : 19146147
Ngành : Kỹ thuật Cơ Điện Tử
Giảng viên hướng dẫn : PGS. TS NGUYỄN TRƯỜNG THỊNH

Tp. Thủ Đức, ngày 25 tháng 12 năm 2021

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH ẢNH	3
DANH MỤC BẢNG	4
CHƯƠNG 1: GIỚI THIỆU	5
1.1. Đặt vấn đề	5
1.2. Giới thiệu đề tài	5
1.3. Ứng dụng của mô hình	5
1.4. Nhiệm vụ thực hiện trong đề tài	6
1.5. Bố cục bài báo cáo	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	7
2.1. Khái niệm xử lý ngôn ngữ tự nhiên	7
2.2. Khái quát về phân loại văn bản	7
2.3. Quy trình xây dựng mô hình cho bài toán phân loại	8
2.4. Phương pháp tách từ	9
2.5. Các phương pháp phân loại văn bản	10
2.6. Các phương pháp đánh giá mô hình	11
CHƯƠNG 3: MÔ HÌNH VÀ GIAO DIỆN	14
3.1. Xây dựng mô hình	14
3.1.1. Chuẩn bị tập dữ liệu	14
3.1.2. Tiền xử lý dữ liệu	14
3.1.3. Vector hoá từ	19
3.1.4. Huấn luyện mô hình bằng mạng Neural Network	21
3.2. Xây dựng giao diện	22
3.2.1. Thiết kế giao diện web	22
3.2.2. Tích hợp mô hình đã huấn luyện lên web	23
CHƯƠNG 4: ĐÁNH GIÁ MÔ HÌNH	24
4.1. Đánh giá mô hình	24
4.2. Thử nghiệm mô hình	26
4.3. Thử nghiệm mô hình theo thời gian thực	27

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	29
5.1. Kết luận	29
5.2. Hướng phát triển	29
TÀI LIỆU THAM KHẢO	31

DANH MỤC HÌNH ẢNH

Hình 1. Khái quát phân loại văn bản	8
Hình 2. Quy trình xây dựng mô hình phân loại văn bản	9
Hình 3. Mô hình Neural Network	11
Hình 4. Bảng ma trận nhầm lẫn.....	12
Hình 5. Đồ thị tập dữ liệu	19
Hình 6. Xây dựng mô hình Neural Network.....	21
Hình 7. Giao diện web	23
Hình 8. Thư viện Flask Python	23
Hình 9. Đồ thị thể hiện quá trình huấn luyện.....	24
Hình 10. Dự đoán theo thời gian thực	28

DANH MỤC BẢNG

Bảng 1. Số lượng dữ liệu được sử dụng trong mô hình	14
Bảng 2. Bảng ma trận nhầm lẫn đánh giá hiệu quả mô hình	25
Bảng 3. Bảng thông số Precision, Recall và F1-score đánh giá mô hình	25

CHƯƠNG 1: GIỚI THIỆU

1.1. Đặt vấn đề

Trong thời đại phát triển ngày nay, sự bùng nổ của mạng Internet đã làm cho không gian dữ liệu mạng gia tăng mạnh, với lượng thông tin được đưa lên mạng ngày càng nhiều nên dữ liệu mạng đã trở thành một kho tài liệu khổng lồ về mọi lĩnh vực và nội dung. Chúng ta có thể truy cập và tìm kiếm các nguồn tài liệu và thông tin chúng ta cần mà không phải tốn kém quá nhiều chi phí.

Tuy tiện lợi nhưng nó cũng có khó khăn trong việc triển khai một lượng lớn thông tin trên các trang mạng. Một trong những khó khăn đó đó là tần suất cập nhật của các thông tin quá lớn. Do sự gia tăng của số lượng văn bản cùng với nhu cầu tìm kiếm văn bản cũng tăng theo nên việc phân loại văn bản là điều thực sự cần thiết, nó giúp chúng ta tìm kiếm thông tin một cách nhanh chóng hơn thay vì phải tìm lần lượt từng văn bản và điều này rất tốn thời gian.

Vì vậy trước vấn đề trên để có thể phân loại các văn bản điện tử một cách hiệu quả và nhanh chóng hơn em đã chọn đề tài “Phân loại văn bản tiếng Việt ” để làm đề tài cho môn học này.

1.2. Giới thiệu đề tài

Phân loại văn bản là bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và ngành trí tuệ nhân tạo. Mục tiêu của bài toán là gán nhãn cho các tài liệu văn bản vào các nhóm chủ đề cho trước.

Bằng cách xây dựng các mô hình với thuật toán như: Naïve Bayes, KNN (K-Nearest_Neighbor), cây quyết định (Decision Tree), mạng thần kinh nhân tạo (Neural Network),...Sau đó tiến hành huấn luyện mô hình với các tập dữ liệu đầu vào được chuẩn bị trước và đã tiền xử lý để được một mô hình hoàn chỉnh có thể dự đoán được đầu ra của các dữ liệu văn bản mới.

Trong đề tài này, em chọn mạng thần kinh nhân tạo (Neural Network) để huấn luyện cho mô hình.

1.3. Ứng dụng của mô hình

Song song với sự phát triển của Internet, nhu cầu cập nhật thông tin của con người ngày càng nâng cao, báo điện tử ra đời nhằm cung cấp thông tin nhanh, đầy đủ và chính xác. Báo điện tử cung cấp lượng thông tin lớn về thời sự hằng

ngày thuộc nhiều chủ đề khác nhau, việc phân loại các chủ đề đó thủ công sẽ tốn rất nhiều thời gian. Vì vậy có thể áp dụng mô hình phân loại văn bản ở đây sẽ tăng độ hiệu quả của việc phân loại chủ đề của các bài báo, cũng như giảm được thời gian phân loại.

1.4. Nhiệm vụ thực hiện trong đề tài

Mô hình:

- Chuẩn bị tập dữ liệu
- Tiền xử lý dữ liệu
- Vector hoá từ
- Xây dựng mạng Neural Network và huấn luyện

Giao diện:

- Thiết kế giao diện web
- Tích hợp mô hình đã huấn luyện lên web

1.5. Bố cục bài báo cáo

- **Chương 1. Giới thiệu:** trình bày tổng quát về đề tài
- **Chương 2: Cơ sở lý thuyết:** trình bày lý thuyết trong xử lý ngôn ngữ tự nhiên và bài toán phân loại văn bản
- **Chương 3. Mô hình và giao diện:** trình bày các bước thực hiện xây dựng mô hình và giao diện
- **Chương 4. Đánh giá mô hình:** thực nghiệm và đánh giá mô hình
- **Chương 5. Kết luận và phát triển:** tổng kết về đề tài và nêu hướng phát triển

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Khái niệm xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên là một nhánh nghiên cứu của trí tuệ nhân tạo, được phát triển nhằm xây dựng các chương trình máy tính có khả năng phân tích, xử lý, và hiểu ngôn ngữ con người.

Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

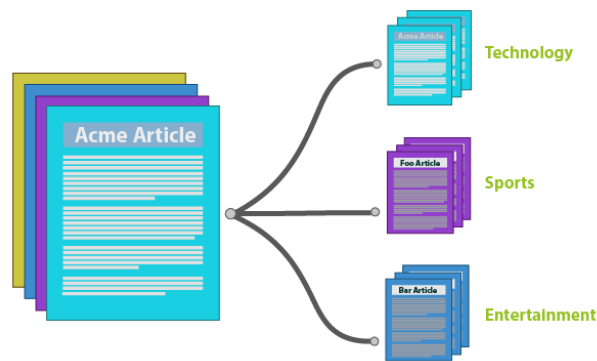
Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing).

- Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh).
- Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, phân loại văn bản.

Tuy nó mạng lại nhiều hiệu quả và được ứng dụng rộng rãi nhưng việc xử lý ngôn ngữ tự nhiên là một nhánh tương đối khó trong ngành trí tuệ nhân tạo vì tập từ điển rộng lớn và được cập nhật thường xuyên, cấu trúc ngữ pháp linh hoạt và đôi khi khá lỏng lẻo, ngôn ngữ đôi khi thể hiện cảm xúc, ẩn ý của người viết.

2.2. Khái quát về phân loại văn bản

Phân loại văn bản là một bài toán học có giám sát, bài toán có nhiệm vụ gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện. Các ứng dụng của phân loại văn bản rất đa dạng như: Hiểu được ý nghĩa, đánh giá, bình luận của người dùng; Lọc email rác; Phân tích cảm xúc; Phân loại tin tức, các bài báo điện tử... Việc tự động phân loại văn bản vào một chủ đề nào đó giúp cho việc sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn về sau.



Hình 1. Khái quát phân loại văn bản

Để giải quyết một bài toán phân loại văn bản, thường trải qua các giai đoạn:

- Chuẩn bị dữ liệu (Data Preparation)
- Tiền xử lý dữ liệu
- Trích chọn đặc trưng (Feature Engineering)
- Xây dựng mô hình phân loại (Build Model)

Trên thế giới đã có nhiều công trình nghiên cứu đạt những kết quả khả quan, nhất là đối với phân loại văn bản tiếng Anh. Tuy vậy, các nghiên cứu và ứng dụng đối với văn bản tiếng Việt còn nhiều hạn chế do khó khăn về tách từ và câu.

2.3. Quy trình xây dựng mô hình cho bài toán phân loại

Đầu vào của quá trình huấn luyện là các dữ liệu văn bản và các nhãn tương ứng với chủ đề cần phân loại. Quá trình này gồm 3 bước: tiền xử lý văn bản, trích xuất đặc trưng và huấn luyện sử dụng các thuật toán học máy. Bước tiền xử lý văn bản gồm 4 công đoạn:

Bước 1: Thực hiện làm sạch dữ liệu để loại bỏ tạp nhiễu nhằm có kết quả xử lý dữ liệu tốt. Đa phần tạp nhiễu là các thẻ HTML, JavaScript.

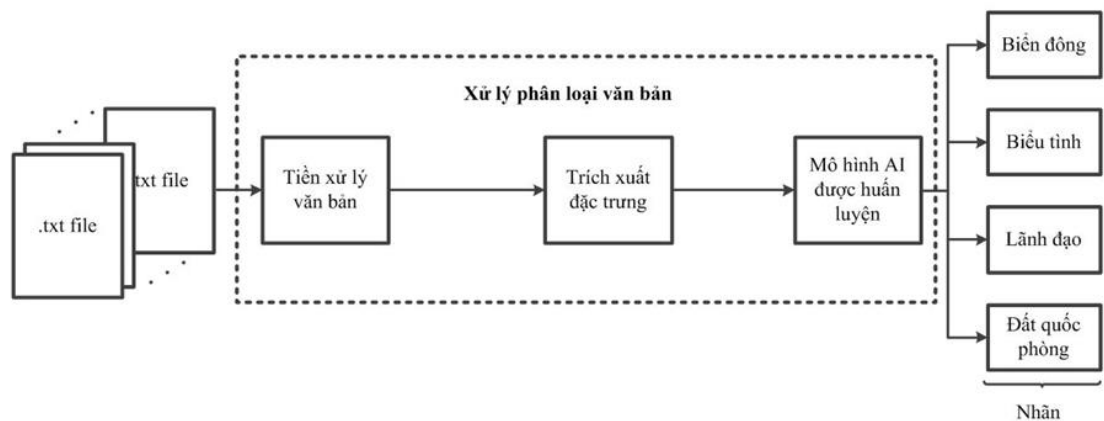
Bước 2: Thực hiện tách từ - một công đoạn quan trọng bậc nhất trong xử lý ngôn ngữ tự nhiên, do Tiếng Việt có độ phức tạp hơn ngôn ngữ khác (bởi có các từ ghép). Việc tách từ theo nhiều cách khác nhau có thể gây ra sự hiểu nhầm về mặt ngữ nghĩa.

Bước 3: Chuẩn hóa từ để đưa văn bản từ các dạng không đồng nhất về cùng một dạng (ví dụ tất cả đều chuẩn về chữ thường). Việc tối ưu bộ nhớ lưu trữ và tính chính xác cũng rất quan trọng. Có nhiều cách viết, mỗi cách viết khi lưu trữ sẽ tốn dung lượng bộ nhớ khác nhau nên tùy theo nhu cầu, tình hình thực tế để đưa văn bản về một dạng đồng nhất.

Bước 4: Loại bỏ những từ không có ý nghĩa (stop words) mà xuất hiện nhiều trong ngôn ngữ tự nhiên. Có 2 cách chính để loại bỏ stop words, đó là dùng từ điển hoặc dựa theo tần suất xuất hiện.

Bước 5: Trích xuất đặc trưng gồm 2 bước là xây dựng từ điển và tạo vector số cho các văn bản theo phương pháp túi đựng từ (Bag of word - BoW). Tất cả các từ trong văn bản cần được chuyển thành dạng biểu diễn số. Cách đơn giản nhất là xây dựng một bộ từ điển, sau đó thay thế từ đó bằng thứ tự xuất hiện trong từ điển để tạo thành một vector ma trận các từ trong câu.

Bước 6: Xây dựng mô hình các thuật toán học máy sẽ huấn luyện một bộ phân loại bằng sử dụng các vector thuộc tính của dữ liệu ở trên đã được gán nhãn.



Hình 2. Quy trình xây dựng mô hình phân loại văn bản

2.4. Phương pháp tách từ

Đối với tiếng Anh từ là các ký tự có nghĩa được tách nhau bởi khoảng trắng trong câu, do vậy việc tách từ trở nên đơn giản. Nhưng đối với tiếng Việt, các từ có nghĩa không chỉ là những từ được phân bởi khoảng trắng, có nhiều từ phải ghép các từ liên kề với nhau mới trở thành một từ có ý nghĩa và nó còn tùy thuộc ngữ cảnh của từng câu. Tuy khó nhưng không thể bỏ qua bước tách từ vì nó sẽ ảnh hưởng nhiều tới kết quả dự đoán của mô hình. Vì vậy nó trở thành một vấn đề thách thức đối với các mô hình xử lý ngôn ngữ tiếng Việt.

Các cách tiếp cận bài toán tách từ:

- Tiếp cận dựa vào từ điển
- Tiếp cận dựa vào thống kê
- Kết hợp từ điển và thống kê

Một số phương pháp tách từ có thể áp dụng cho ngôn ngữ tiếng Việt như:

- So khớp dài nhất (Longest Matching)
- So khớp cực đại (Maximum Matching)
- Mô hình Markov ẩn (Hidden Markov Models – HMM)
- Độ hỗn loạn cực đại (Maximum Entropy – ME)
- Máy học sử dụng vector hỗ trợ (Support Vectors Machines)
- Ngoài ra còn nhiều phương pháp khác

Hiện nay tách từ tiếng Việt có các bộ công cụ mã nguồn mở hỗ trợ trong việc xử lý ngôn ngữ tiếng Việt được phát triển như Underthesea, Pyvi. Các thư viện được hỗ trợ đầy đủ các chức năng trong việc xử lý ngôn ngữ từ xóa dấu, xóa ký tự đặc biệt và tách từ với độ chính xác cao.

Trong bài báo cao này em sẽ tiếp cận bài toán tách từ bằng cách tiếp cận cổ điển là sử dụng bộ từ điển tiếng Việt có sẵn và phương pháp so khớp dài nhất.

2.5. Các phương pháp phân loại văn bản

Phân loại văn bản là một lĩnh vực quan trọng trong việc quản lý các văn bản, có nhiều phương pháp được sử dụng nhất là đối với ngôn ngữ tiếng Anh. Các phương pháp được áp dụng thành công được sử dụng phổ biến như:

- Mô hình láng giềng gần nhất (K-nearest neighbor - KNN)

KNN là phương pháp truyền thống của lĩnh vực học máy được áp dụng cho nhiều bài toán khác nhau, KNN tiếp cận theo phương pháp thống kê dữ liệu. Khi thực hiện dự đoán một dữ liệu mới thì thuật toán sẽ tính khoảng cách của tất cả các dữ liệu trong tập huấn luyện đến điểm dữ liệu mới này để tìm ra K điểm dữ liệu gần nhất, sau đó dùng khoảng cách đó đánh trọng số cho tất cả các lớp ngõ ra, dữ liệu nào không thuộc K điểm dữ liệu sẽ có trọng số bằng 0. Trong K điểm dữ liệu, dữ liệu nào có trọng số cao nhất sẽ chọn lớp ngõ ra của dữ liệu đó làm ngõ ra cho dữ liệu mới.

- Mô hình xác suất Naive Bayes

Naive Bayes là phương pháp phân loại dựa vào xác suất được sử dụng rộng rãi trong học máy được sử dụng lần đầu tiên vào năm 1961. Phương pháp này sẽ sử dụng xác suất có điều kiện giữa dữ liệu và phân lớp ngõ ra của dữ liệu để dự đoán xác suất phân lớp ngõ ra của điểm dữ liệu cần phân loại.

- Mô hình máy học vector hỗ trợ (Support Vector Machine - SVM)

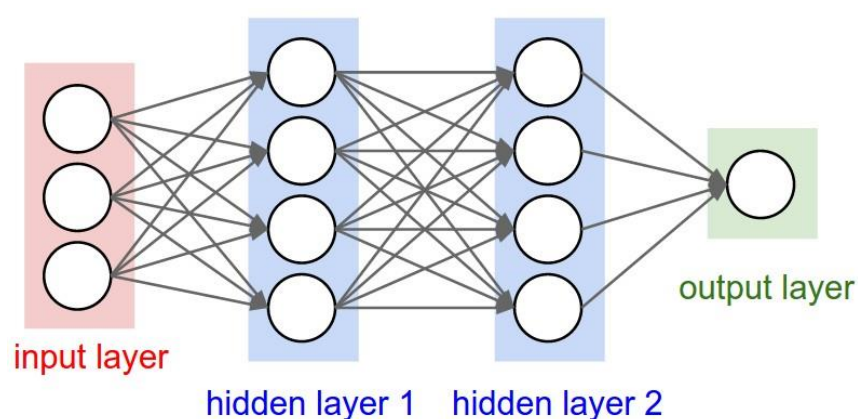
SVM là phương pháp tiếp cận phân loại rất hiệu quả của Vapnik được giới thiệu vào năm 1995. Phương pháp này tìm sẽ tìm ra một mặt phẳng H quyết định tốt nhất có thể chia các điểm trên không gian thành hai lớp riêng biệt. Chất lượng của mô hình phụ thuộc vào khoảng cách từ mặt phẳng được chọn đến điểm dữ liệu gần nhất của mỗi lớp, khoảng cách càng lớn thì khả năng phân loại càng tốt.

- Mạng Neural Network

Neural Network hay còn gọi là Mạng neural nhân tạo là mạng sử dụng các mô hình toán học phức tạp để xử lý thông tin. Chúng dựa trên mô hình hoạt động của các tế bào thần kinh và khớp thần kinh trong não của con người. Tương tự như bộ não con người, mạng nơ-ron nhân tạo kết nối các nút đơn giản, còn được gọi là tế bào thần kinh. Và một tập hợp các nút như vậy tạo thành một mạng lưới các nút, do đó có tên là mạng nơ-ron nhân tạo.

Một mạng Neural Network có 3 thành phần bao gồm:

- Lớp đầu vào đại diện cho các dữ liệu đầu vào.
- Lớp ẩn đại diện cho các nút trung gian phân chia không gian đầu vào thành các vùng có ranh giới.
- Lớp đầu ra đại diện cho đầu ra của mạng neural.



Hình 3. Mô hình Neural Network

2.6. Các phương pháp đánh giá mô hình

Sau khi xây dựng thành công mô hình machine learning và huấn luyện nó trên tập dữ liệu, tiếp theo ta sẽ thực hiện đánh giá hiệu quả của mô hình. Qua đó có

thể biết được liệu mô hình của ta có thành công hay không, mức độ hiệu quả của mô hình đến mức nào,...Từ đó sẽ đưa ra các kết luận, điều chỉnh lại mô hình hoặc dừng việc huấn luyện mô hình.

Các phương pháp đánh giá mô hình phân loại bao gồm:

- **Confusion matrix (ma trận nhầm lẫn):** là một kỹ thuật đánh giá hiệu năng của mô hình cho các bài toán phân lớp. Confusion matrix là một ma trận thể hiện số lượng điểm dữ liệu thuộc vào một class và được dự đoán thuộc vào class.

Confusion matrix cung cấp thêm thông tin về tỉ lệ phân lớp đúng giữa các lớp, hay giúp phát hiện các lớp có tỉ lệ phân lớp nhầm cao nhờ vào các khái niệm True (False) Positive (Negative).

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Hình 4. Bảng ma trận nhầm lẫn

Trong đó:

- True Positive (TP): là số ca dự đoán dương tính đúng.
 - True Negative (TN): là số ca dự đoán dương tính sai.
 - False Positive (FP): là số ca dự đoán âm tính đúng.
 - False Negative (FN): là số ca dự đoán âm tính sai.
- **Accuracy (độ chính xác):** là phương pháp đánh giá mô hình dự đoán thường xuyên đến mức nào. Độ chính xác là tỉ lệ giữa các dữ liệu được dự đoán đúng trên tổng số dữ liệu được đưa vào dự đoán.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** để đo lường dự đoán các điểm dữ liệu được mô hình phân loại vào lớp Positive thì có bao nhiêu dữ liệu thực sự thuộc nhóm Positive.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** để cho biết có bao nhiêu điểm dữ liệu thực sự ở lớp Positive được mô hình phân loại đúng trên tổng các điểm dữ liệu thực sự là Positive.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score:** là trung bình điều hoà giữa Precision và Recall. Do hai giá trị Precision và Recall thường không cân bằng nhau nên không thể đánh giá được chính xác chất lượng mô hình. Vì vậy để đánh giá cả hai giá trị cùng lúc ta sử dụng f1-score.

$$F1 - score = 2 \frac{Precision \times Recall}{Precision + Recall}$$

CHƯƠNG 3: MÔ HÌNH VÀ GIAO DIỆN

3.1. Xây dựng mô hình

3.1.1. Chuẩn bị tập dữ liệu

Trong đề tài này em sử dụng tập dataset có sẵn gồm 10 chủ đề bài báo với các lĩnh vực: Chính trị Xã hội, Đời sống, Khoa học, Kinh doanh, Pháp Luật, Sức Khỏe, Thế giới, Thể thao, Văn hoá và Công nghệ. Các dữ liệu được lưu dưới dạng file txt.

Bảng 1. Số lượng dữ liệu được sử dụng trong mô hình

Số thứ tự	Tên chủ đề	Số lượng
0	Chính trị Xã hội	4,472
1	Công nghệ	4,173
2	Đời sống	4,365
3	Khoa học	3,916
4	Kinh doanh	4,327
5	Pháp luật	4,345
6	Sức khỏe	4,483
7	Thế giới	4,307
8	Thể thao	4,287
9	Văn hoá	4,300
Tổng		42,975

3.1.2. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là giai đoạn rất quan trọng, hay nói cách khác đây là công đoạn làm sạch văn bản. Việc văn bản được làm sạch giúp thuật toán có thể trích xuất được những đặc trưng tốt nhất từ đó nâng cao hiệu quả, chất lượng của các mô hình. Ứng với mỗi đoạn văn bản, em tiến hành loại bỏ các ký hiệu và các chữ số không cần thiết, sau đó thực hiện tách từ có nghĩa, loại bỏ các từ dừng và cuối cùng là xoá dấu câu. Các bước thực hiện này nhằm giảm chiều dữ liệu làm tăng tốc độ và hiệu quả huấn luyện cho mô hình.

Để thực hiện tiền xử lý tập dữ liệu thô đầu tiên em tạo các hàm con xử lý sau đây:

- Hàm xoá các ký hiệu:

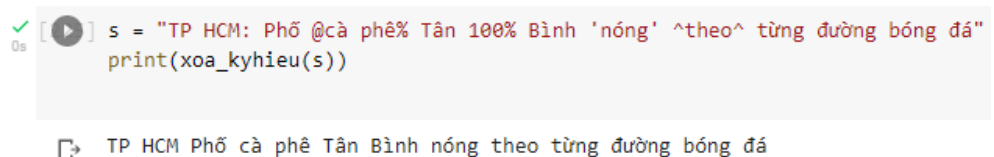
Các ký hiệu đặc biệt và các số không mang yếu tố phân loại văn bản thuộc chủ đề nào và tần số xuất hiện của chúng cũng khá nhiều nên việc loại bỏ các ký tự này là điều cần thiết.

```
special_character = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '=', '+', '\\n', '\\t', ':', ';', ',', '.', '|', "'", '/', '\\']

def xoa_kyhieu(s):
    b = []
    for word in s.split():
        a = []
        for letter in word:
            if letter not in special_character:
                a.append(letter)
        mystring = "".join([str(char) for char in a])
        if mystring != "":
            b.append(mystring)
    mystringfinal = " ".join([str(char) for char in b])
    return mystringfinal
```

Bằng cách tạo ra một list các ký hiệu đặc biệt muốn loại bỏ sau đó ta xét từng thành phần nhỏ nhất trong đoạn văn bản và nếu thành phần nào có trong list ký tự đặc biệt thì loại bỏ nó sau đó ghép lại thành câu như cũ.

Kết quả:



```
s = "TP HCM: Phố @cà phê Tân 100% Bình 'nóng' ^theo^ từng đường bóng đá"
print(xoa_kyhieu(s))
```

TP HCM Phố cà phê Tân Bình nóng theo từng đường bóng đá

- Hàm tách từ:

Việc phân loại tập tin được thực hiện dựa trên đặc trưng của tập tin được cấu tạo nên từ các từ và cụm từ. Vì vậy quá trình phân loại có tốt hay không sẽ phụ thuộc rất lớn vào quá trình tách từ.

```
dir = '/content/drive/MyDrive/Text_classification/Viet74K.txt'
file = open(dir, 'r', encoding='utf-8')
```

```
viet_dict = dict()

for word in file:
    word = re.sub(r'[\n]', '', word)
    if word not in viet_dict:
        viet_dict[word] = len(viet_dict)

def tachtu(text, dict):
    input = text.split(" ")
    words = []
    start = 0
    while True:
        end = len(input)
        while end > start: #e>s khi còn từ
            sentence = input[start:end]
            sentence = " ".join(sentence)
            end = end-1
            if sentence.lower() in viet_dict:
                words.append(sentence)
                break
        start = end + 1
        if start == len(input):
            break
    output = []
    for word in words:
        word = re.sub(r'[ ]', '_', word)
        output.append(word)
    output = " ".join(output)
    return output
```

Khác với tiếng Anh, trong tiếng Việt dấu cách không mang ý nghĩa phân tách một từ có nghĩa mà chỉ là tách các âm tiết với nhau nên tách từ theo khoảng trắng sẽ không giữ lại được đầy đủ ý nghĩa của từ. Trong mô hình này em tạo một hàm tách từ theo phương pháp Longest mactching (So khớp dài nhất) bằng cách sử dụng một bộ từ điển tiếng Việt để so sánh, bắt đầu xét các câu muốn tách từ và xét cụm từ dài nhất trong câu, sau đó giảm dần cho đến khi xuất hiện từ tương đương trong từ điển thì tách nó ra làm một từ. Tiếp tục thực hiện xét lại với cụm từ dài nhất mới cho đến khi xét hết các từ trong câu.

Kết quả:

☞ Phố cà_phê Tân Bình nóng theo từng đường bóng_đá

- Hàm xoá dấu:

Việc xóa dấu câu có thể gây mất nghĩa của câu, tuy nhiên có thể làm giảm chiều dài vector từ cho đoạn văn bản.

```
def xoa_dau(s) :  
    s = re.sub(r'[àáâãäåăąǎȁǻǿ]', 'a', s)  
    s = re.sub(r'[\AA\B\BB\CC\DD\EE\FF\G\GG]', 'A', s)  
    s = re.sub(r'[èéêëēĕěĉĥ]', 'e', s)  
    s = re.sub(r'[ÊËẼỄỀẾỂỄ]', 'E', s)  
    s = re.sub(r'[òóôõöōŏōðōōøṽ]', 'o', s)  
    s = re.sub(r'[ÓÒÔÕÖŌŐŐỖỠỖỖ]', 'O', s)  
    s = re.sub(r'[ìíîïĩ]', 'i', s)  
    s = re.sub(r'[ÍÎỊĨĨ]', 'I', s)  
    s = re.sub(r'[ùúûüũừứựữ]', 'u', s)  
    s = re.sub(r'[UÜÚÝÛỦỪỨỰỮ]', 'U', s)  
    s = re.sub(r'[ỳýỵỷỹ]', 'y', s)  
    s = re.sub(r'[ỚỤờỞỜ]', 'Y', s)  
    s = re.sub(r'[Đ]', 'D', s)  
    s = re.sub(r'[đ]', 'd', s)  
  
    return s
```

Để loại bỏ dấu câu em sử dụng lệnh sub của thư viện re, bằng lệnh sub nó sẽ dò theo câu văn bản được đưa vào và từ nào giống với từ trong [] sẽ được thay thế bằng từ trong nháy đơn.

Kết quả:

TP HCM Pho ca_phe Tan_Binh nong theo tung duong bong_da

- Loại bỏ Stopword:

```
stopword = ['a_lô']

def create_stopword(path):
    with open(path, encoding="utf-8") as words:
        return [w[:len(w) - 1] for w in words] + stopword

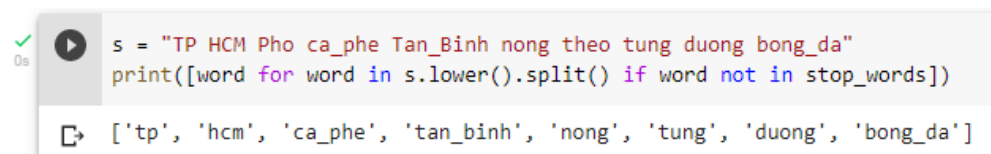
stop_words = create_stopword('/content/drive/MyDrive/Text classification/stop word.txt')
```

```
s = "TP HCM Pho ca_phe Tan_Binh nong theo tung duong bong_da"
print([word for word in s.lower().split() if word not in stop_words])
```

Stopword là các từ xuất hiện nhiều lần và ở hầu hết các thể loại, nó sẽ gây nhiễu trong quá trình phân loại. Do đó những từ này không có ý nghĩa trong việc phân loại. Vì vậy cần loại bỏ các từ này nhằm giảm chiều của vector khi thực hiện vector hoá một đoạn văn bản làm tăng hiệu quả huấn luyện, tăng độ chính xác cho phân loại và giảm thời gian huấn luyện mô hình.

Để thực hiện loại bỏ các Stopword đầu tiên cần xác định các từ xuất hiện nhiều lần, thường được xét trong một văn bản dài. Trong mô hình này em sử dụng một danh sách Stopword có sẵn.

Kết quả:



```
s = "TP HCM Pho ca_phe Tan_Binh nong theo tung duong bong_da"
print([word for word in s.lower().split() if word not in stop_words])
```

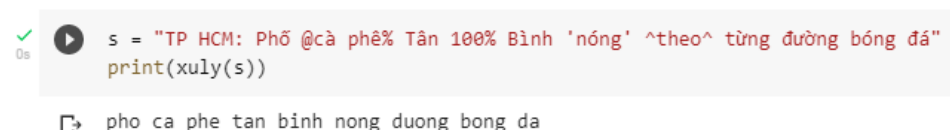
['tp', 'hcm', 'ca_phe', 'tan_binh', 'nong', 'tung', 'duong', 'bong_da']

- Hàm xử lý:

Hàm này chỉ chứa các hàm con khác để khi dùng lệnh thì việc nhập lệnh được đơn giản hơn.

```
def xuly(s):
    s = xoa_kyhieu(s)
    s = tachtu(s, viet_dict)
    s = [word for word in s.lower().split() if word not in stop_words]
    s = " ".join(s)
    s = xoa_dau(s)
    return s
```

Kết quả:



```
s = "TP HCM: Phố @cà phê% Tân 100% Bình 'nóng' ^theo^ từng đường bóng đá"
print(xuly(s))
```

pho ca_phe tan binh nong duong bong_da

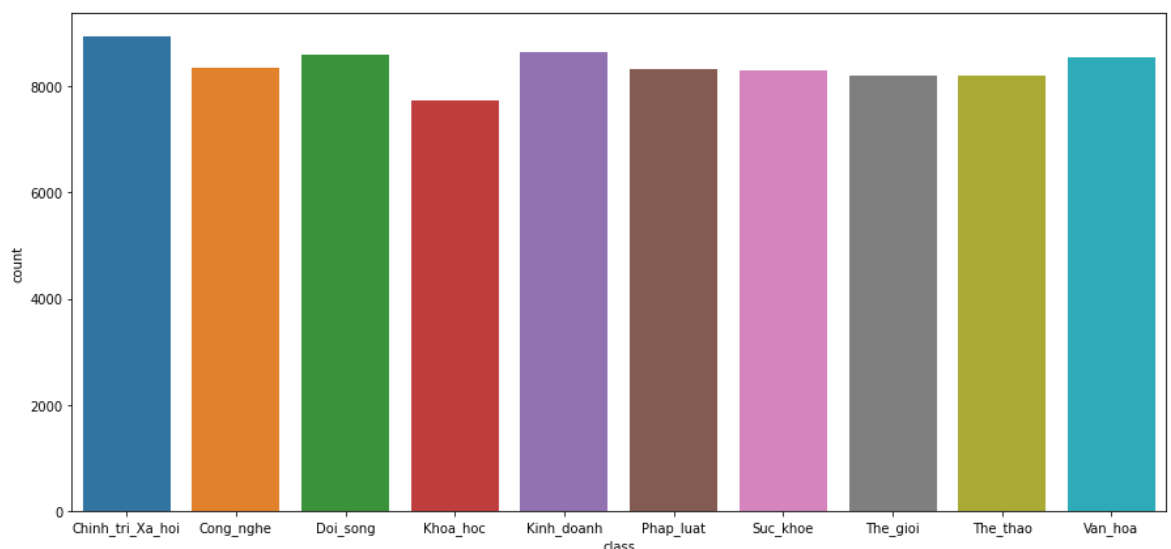
Sau khi đã có hàm tiền xử lý cuối cùng ta thực hiện đọc các file txt trong tập dữ liệu đã chuẩn bị sẵn. Với mỗi file txt, em thực hiện đọc 2 lần, mỗi lần đọc 2 dòng sau đó ghép lại và gán nhãn cho dữ liệu đó.

```

data = pd.DataFrame({'content':[], 'class': []})
dir = '/content/drive/MyDrive/Text_classification/dataset'
for i in os.listdir(dir):
    for j in os.listdir(dir+ '/' +i):
        file = open(dir+ '/' + i + '/' + j, 'r', encoding='utf-16')
        for k in range(2):
            d1 = file.readline()
            d2 = file.readline()
            d = xuly(d1+d2)
            if d != "":
                data = data.append({'content': d, 'class':i}, ignore_index=True)

```

Sau khi hoàn tất việc tiền xử lý dữ liệu ta thu được một bảng dữ liệu khoảng hơn 80.000 dữ liệu thuộc 10 chủ đề khác nhau. Mỗi chủ đề trung bình là 8.000 dữ liệu không quá mất cân bằng dữ liệu



Hình 5. Đồ thị tập dữ liệu

3.1.3. Vector hoá từ

Để có thể huấn luyện mô hình ta cần chuyển đổi các dữ liệu thành dạng vector số mà máy tính có thể hiểu, vì vậy đầu tiên ta cần vector hoá các dữ liệu đặc trưng đầu vào và chủ đề đầu ra của dữ liệu đó.

Để vector hoá ta tạo ra một từ điển chứa các từ, sau đó xét các từ trong dữ liệu đã qua xử lý và gán các trọng số cho các từ xuất hiện.

```

word_dict = dict()
max_num = 0

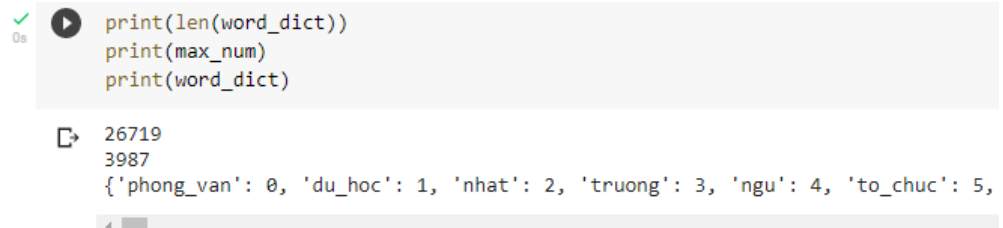
for sentence in input_s:

```

```
for word in sentence:
    if word not in word_dict:
        word_dict[word] = len(word_dict)
if len(sentence) > max_num:
    max_num = len(sentence)
```

Để thực hiện ta tạo một từ điển rỗng, sau đó tiến hành xét từng từ trong mỗi câu nếu từ nào không có trong từ điển thì thêm vào và gán trọng số tương ứng với chiều dài của từ điển khi đó. Tiếp tục xét cho đến khi hết từ trong tập dữ liệu.

Kết quả:



```
print(len(word_dict))
print(max_num)
print(word_dict)
```

```
26719
3987
{'phong_van': 0, 'du_hoc': 1, 'nhat': 2, 'truong': 3, 'ngu': 4, 'to_chuc': 5,
```

Kết quả thu được là một bộ từ điển chứa các từ đã được xử lý, với độ dài của từ điển là 26.719 từ. Trong đó câu được xét có độ dài lớn nhất là 3987 từ.

Sau khi đã tạo được từ điển ta bắt đầu vector hoá các câu trong tập dữ liệu. Đối với dữ liệu đầu vào là các câu văn bản ta xét từng từ trong câu và so sánh với các từ trong từ điển, nếu từ trong câu tương đương với từ trong từ điển thì gán từ đó bằng trọng số của từ tương đương trong từ điển. Đồng thời tăng độ dài của vector bằng cách thêm các phần tử 0 sao cho độ dài của một vector bằng với độ dài của câu dài nhất trong tập dữ liệu. Việc này giúp độ dài của tất cả các vector đặc trưng là như nhau, sẽ dễ dàng hơn trong việc huấn luyện.

Đối với dữ liệu đầu ra là các chủ đề ta thực hiện one-hot encoding để tạo các vector mà với mỗi chủ đề sẽ được đại diện với một vector khác nhau và duy nhất.

```
X = []
Y = []
X = [[word_dict[word] for word in line] for line in input_
s]
X = sequence.pad_sequences(X, max_num)
Y = data['class']
label_encode = LabelEncoder()
```

```
Y = label_encode.fit_transform(Y)
Y_train = to_categorical(Y, num_classes=10)
```

Kết quả:

```

✓ 0s ▶ print(X.shape)
      print(Y_train.shape)
      print(X[5])
      print(Y_train[5])

↳ (83797, 3987)
   (83797, 10)
   [ 0  0  0 ... 90 91 92]
   [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

```

3.1.4. Huấn luyện mô hình bằng mạng Neural Network

Sau khi đã xử lý các dữ liệu đầu vào và chuyển đổi thành các vector, tiếp theo ta sẽ xây dựng mô hình Neural để huấn luyện dữ liệu.

Với các dữ liệu đã được vector hoá ta chia thành hai phần train và test với tỉ lệ là 75% dữ liệu cho train và 15% dữ liệu cho test.

Sau đó ta xây dựng mô hình với đầu vào là sẽ qua một lớp Embedding, sau đó qua một lớp GlobalAveragePooling1D, Dropout và đầu ra là 10 ngõ ra.

```

from keras.models import Sequential
from keras.layers import Dense, Embedding, Dropout, GlobalAveragePooling1D

model = Sequential()
model.add(Embedding(len(word_dict), 300, input_length=max_num))
model.add(GlobalAveragePooling1D())
model.add(Dropout(0.2))
model.add(Dense(10, activation="softmax"))
model.summary()

```

↳ Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 3987, 300)	8015700
global_average_pooling1d_1 (GlobalAveragePooling1D)	(None, 300)	0
dropout_1 (Dropout)	(None, 300)	0
dense_1 (Dense)	(None, 10)	3010

```

Total params: 8,018,710
Trainable params: 8,018,710
Non-trainable params: 0

```

Hình 6. Xây dựng mô hình Neural Network

Mô hình được biên dịch với 3 thành phần cơ bản là loss (hàm mất mát), Optimizer (trình tối ưu hoá), Metrics (phương pháp đánh giá). Trong đó:

- hàm Loss được sử dụng để tìm lỗi hoặc sai lệch trong quá trình học. Trong mô hình em chọn hàm mất mát là “Categorical_crossentropy”.
- Trình tối ưu hóa là một quá trình quan trọng nhằm tối ưu các trọng số đầu vào bằng cách so sánh dự đoán và hàm mất mát. Trong mô hình em chọn trình tối ưu hoá là “Adam”.
- Phương pháp đánh giá Metrics được sử dụng để đánh giá hiệu suất của mô hình. Trong mô hình em chọn đánh giá mô hình theo độ chính xác “Accuracy”

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])
```

Sau khi biên dịch mô hình ta tiến hành thực hiện huấn luyện mô hình với tổng 30 lần huấn luyện. Trong mỗi lần huấn luyện, mô hình sẽ gom lần lượt số dữ liệu theo batch_size để huấn luyện.

```
history = model.fit(X_train,y_train,batch_size=128,epochs=30,validation_data=(X_test,y_test),verbose=1)
```

3.2. Xây dựng giao diện

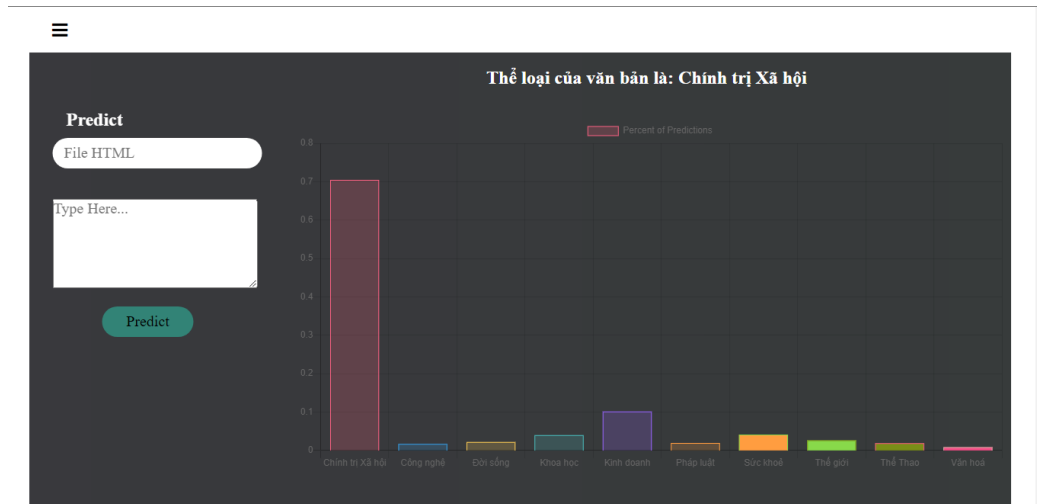
3.2.1. Thiết kế giao diện web

Giao diện được em thiết kế bằng ngôn ngữ web HTML (Hypertext Markup Language), được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles,...

Giao diện em thiết kế gồm có hai phần:

Phần thứ nhất là các ô văn bản để nhập văn bản hoặc thêm đường dẫn đến các trang báo điện tử, khi các văn bản hoặc đường dẫn trang báo điện tử được thêm vào và nhấn predict thì các dữ liệu đó sẽ được lấy xuống thực hiện các bước tiền xử lý và dự đoán kết quả.

Phần thứ hai là phần thể hiện kết quả dự đoán, chủ đề được dự đoán sẽ được in lên màn hình giao diện đồng thời cũng vẽ lên đồ thị thống kê tỷ lệ dự đoán của các chủ đề.



Hình 7. Giao diện web

3.2.2. Tích hợp mô hình đã huấn luyện lên web

Sau khi đã thiết kế được giao diện ta sử dụng thư viện Flask của ngôn ngữ Python để tích hợp mô hình deep learning đã huấn luyện và giao diện web lên một server.



Hình 8. Thư viện Flask Python

Thư viện Flask là một framework ứng dụng cho website được tạo ra từ ngôn ngữ lập trình web Python. Công cụ này có dung lượng khá nhẹ nhưng lại rất linh hoạt trong công dụng. Nó được ứng dụng trong thiết kế website đơn giản và tạo lập các ứng dụng cho những trang web lớn và phức tạp.

CHƯƠNG 4: ĐÁNH GIÁ MÔ HÌNH

4.1. Đánh giá mô hình

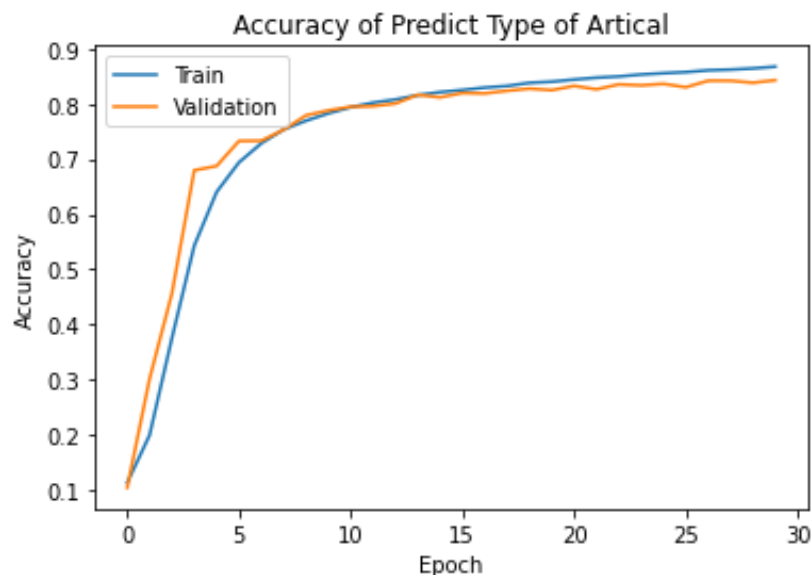
Đánh giá là một quá trình trong quá trình phát triển mô hình để kiểm tra xem liệu mô hình có phù hợp nhất với vấn đề đã cho và dữ liệu tương ứng hay không.

Em thực hiện đánh giá với 12.570 mẫu dữ liệu kiểm tra được chia ra từ tập dữ liệu gốc.

Sau khi hoàn thành huấn luyện mô hình bằng mạng Neural thì độ chính xác của mô hình tương đối khả quan. Trong đó:

Đối với tập huấn luyện: độ chính xác là 86.72% và sai số là 0.45

Đối với tập kiểm tra: độ chính xác là 84.26% và sai số là 0.55



Hình 9. Đồ thị thể hiện quá trình huấn luyện

Qua đồ thị có thể thấy độ chính xác kiểm tra bám sát với độ chính xác huấn luyện, không xuất hiện hiện tượng overfitting. Sau 20 lần huấn luyện thì độ chính xác kiểm tra có xu hướng bão hòa.

- Đánh giá mô hình bằng Confusion matrix:

Bảng 2. Bảng ma trận nhầm lẫn đánh giá hiệu quả mô hình

Tên chủ đề	CH-XH	Công nghệ	Đời sống	Khoa học	Kinh doanh	Pháp luật	Sức khỏe	Thể giới	Thể thao	Văn hoá
CH-XH	962 7.65%	39 0.31%	53 0.42%	27 0.21%	59 0.47%	104 0.83%	15 0.19%	24 0.19%	13 0.1%	25 0.2%
Công nghệ	13 0.1%	1172 9.32%	9 0.07%	14 0.11%	24 0.19%	7 0.06%	2 0.02%	14 0.11%	4 0.03%	13 0.1%
Đời sống	34 0.27%	35 0.28%	920 7.32%	62 0.49%	17 0.14%	23 0.18%	42 0.33%	17 0.14%	19 0.15%	110 0.88%
Khoa học	13 0.1%	53 0.42%	75 0.6%	900 7.16%	15 0.12%	3 0.02%	57 0.45%	31 0.25%	11 0.09%	14 0.11%
Kinh doanh	34 0.27%	63 0.5%	14 0.11%	8 0.06%	1110 8.83%	21 0.17%	3 0.02%	39 0.31%	8 0.06%	3 0.02%
Pháp luật	47 0.37%	16 0.13%	24 0.19%	1 0.01%	28 0.22%	1094 8.7%	5 0.04%	11 0.09%	16 0.13%	9 0.07%
Sức khỏe	10 0.08%	11 0.09%	45 0.36%	54 0.43%	6 0.05%	6 0.05%	1107 8.81%	12 0.1%	5 0.04%	7 0.06%
Thể giới	17 0.14%	25 0.2%	25 0.2%	31 0.25%	29 0.23%	13 0.1%	14 0.11%	1024 8.15%	8 0.06%	19 0.15%
Thể thao	5 0.04%	9 0.07%	14 0.11%	2 0.02%	0 0.0%	9 0.07%	4 0.03%	10 0.08%	1155 9.19%	12 0.1%
Văn hoá	8 0.06%	14 0.11%	65 0.52%	14 0.11%	1 0.01%	6 0.05%	6 0.05%	17 0.14%	5 0.04%	1148 9.13%

Từ bảng đánh giá confusion matrix ở trên ta có thể suy ra các thông số precision, recall và f1-score như sau:

Bảng 3. Bảng thông số Precision, Recall và F1-score đánh giá mô hình

Tên chủ đề	Precision	Recall	F1-score
CH-XH	84%	73%	78%
Công nghệ	82%	92%	87%
Đời sống	74%	72%	73%
Khoa học	81%	77%	79%
Kinh doanh	86%	85%	86%
Pháp luật	85%	87%	86%
Sức khỏe	88%	88%	88%
Thể giới	85%	85%	85%
Thể thao	93%	95%	94%
Văn hoá	84%	89%	87%
Tổng	84%		

Qua các thông số trên ta có thể thấy rằng mô hình có độ chính xác dự đoán cao nhất là chủ đề Thể thao với thông số F1-score là 94%, có thể dự đoán đúng 1155 điểm dữ liệu trên tổng 1220 điểm dữ liệu thực sự thuộc chủ đề Thể thao.

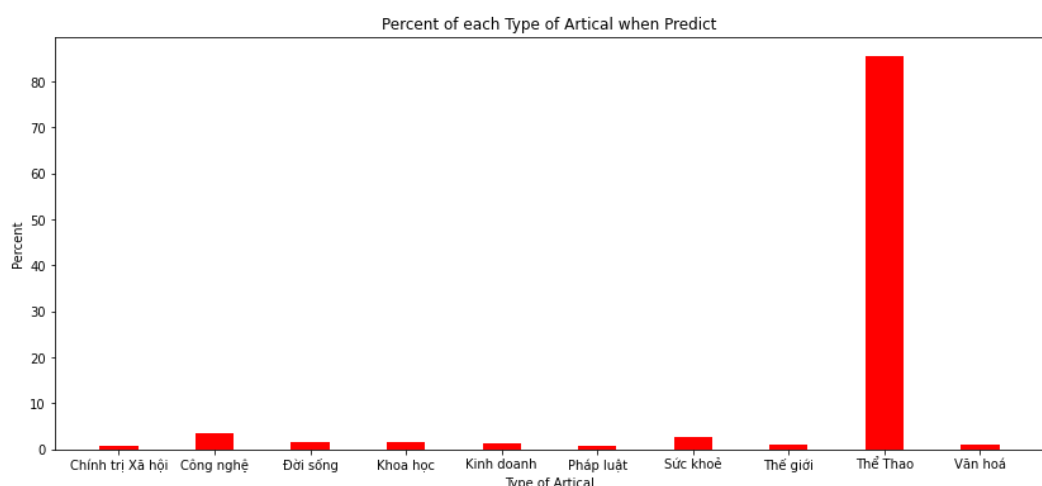
Độ chính xác dự đoán thấp nhất là chủ đề Đời sống với thông số F1-score là 73% và có thể dự đoán đúng 920 điểm dữ liệu trên tổng 1279 điểm dữ liệu thực sự thuộc chủ đề Đời sống

4.2. Thử nghiệm mô hình

Đầu tiên em thực hiện thử nghiệm cho mô hình dự đoán các đoạn văn bản trực tiếp. Với một đoạn văn bản mới, để dự đoán ta thực hiện các bước xử lý cơ bản như xóa ký tự, tách từ, loại bỏ stop word và xóa dấu câu tương tự việc thực hiện xử lý cho tập dữ liệu huấn luyện. Sau đó chuyển hoá đoạn văn bản thành vector đồng thời tăng kích thước vector tương đương với kích thước đầu vào của mô hình. Cuối cùng ta thực hiện dự đoán chủ đề của văn bản mới này.

```
test = "FIFA từ ngày 1/7 sẽ cấm trường hợp như thủ môn Australia  
Andrew Redmayne nhảy khiêu khích tiền đạo Peru ở loạt đá luân l  
ưu vớt World Cup 2022"  
test = xuly(test).split()  
pred = []  
pred.append([word_dict[word] for word in test if word in word_di  
ct])  
pred = sequence.pad_sequences(pred, max_num)  
pred_text = names[np.argmax(model.predict(pred))]  
print(pred_text)
```

Kết quả:



Kết quả dự đoán cho ra chủ đề của đoạn văn bản là thể thao đúng với chủ đề mà nội dung văn bản thể hiện.

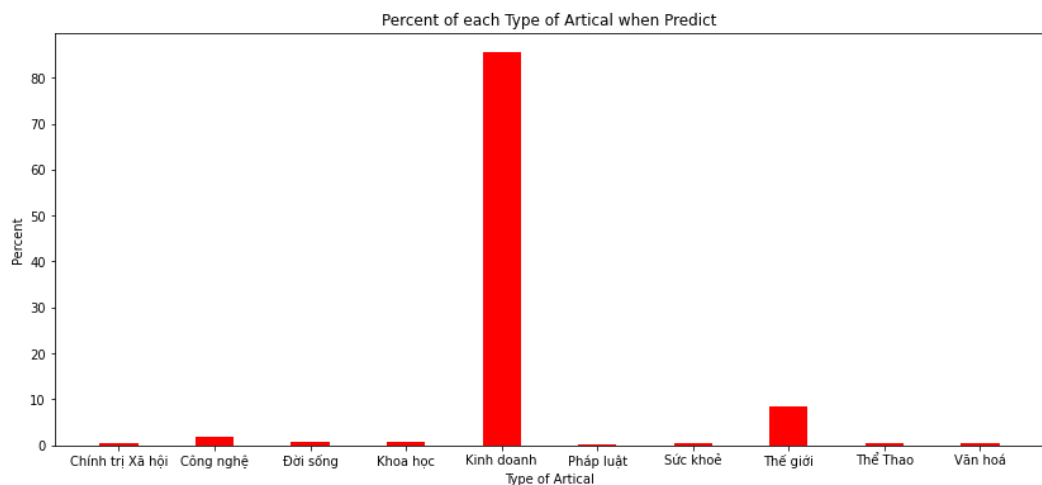
Tiếp theo em thực hiện dự đoán bài báo điện tử thông qua các đường dẫn dạng web. Các đường dẫn đó được xử lý bởi thư viện BeautifulSoup để có thể đọc các nội dung trong trang báo điện tử đó. Trích xuất một phần nội dung của bài báo

và thực hiện tiền xử lý, vector hoá cuối cùng là dự đoán đoạn văn bản được trích xuất, chủ đề của đoạn văn cũng là chủ đề của bài báo.

```
import requests
from bs4 import BeautifulSoup

res = requests.get('https://vnexpress.net/cuoc-chien-chong-lam-
phat-o-cac-nuoc-giau-4476731.html')
res = str(res.text)
soup = BeautifulSoup(res, "html.parser")
title = soup.title.get_text()
content = soup.meta['content']
text = title + ' ' + content
text = xuly(text).split()
pred = []
pred.append([word_dict[word] for word in text if word in word_dict])
pred = sequence.pad_sequences(pred, max_num)
names[np.argmax(model.predict(pred))]
```

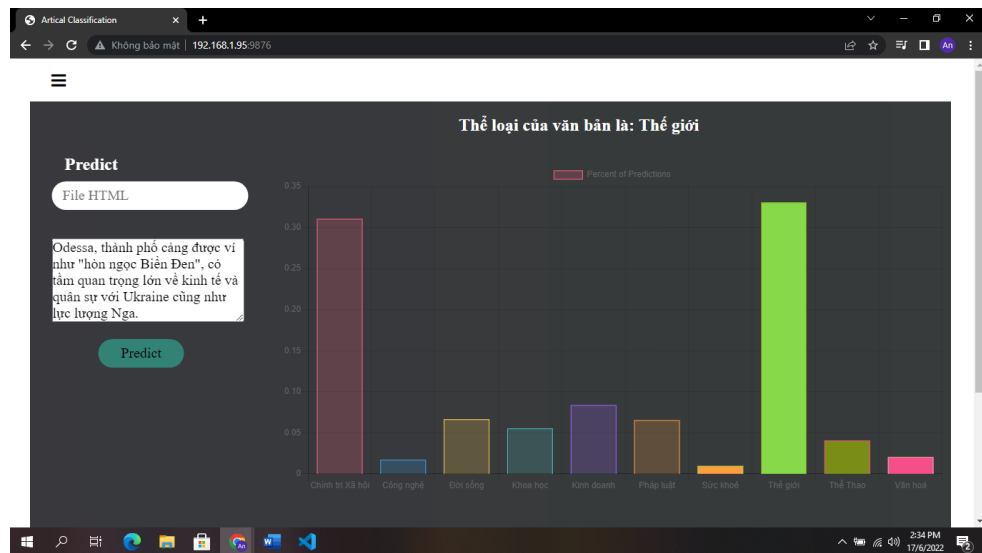
Kết quả:



Kết quả khi thực hiện dự đoán bài báo điện tử bằng đường dẫn cũng cho ra kết quả khá chính xác với chủ đề của bài báo điện tử là Kinh doanh.

4.3. Thử nghiệm mô hình theo thời gian thực

Sau khi đã huấn luyện mô hình ta lưu lại mô hình dưới dạng file (.h5) và tích hợp vào giao diện web bằng thư viện Flask của Python. Việc phân loại cũng tương đối đơn giản, chỉ cần điền đoạn văn bản hoặc một link HTML thì kết quả sẽ được in ra trực tiếp trên giao diện. Đồng thời tỉ lệ dự đoán các chủ đề cũng được vẽ lên.



Hình 10. Dự đoán theo thời gian thực

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Trong nội dung bài báo cáo này em đã nghiên cứu và tìm hiểu về bài toán phân loại văn bản tiếng Việt bằng mô hình Neural Network.

Phân lý thuyết: làm rõ được các khái niệm phân loại văn bản trong xử lý ngôn ngữ, cách tiếp cận bài toán phân loại văn bản, các phương pháp xây dựng mô hình phân loại và đánh giá các mô hình,...

Phân thực hành: xây dựng được các hàm tiền xử lý văn bản cho các dữ liệu đầu vào, vector hoá các dữ liệu bằng từ điển tiếng Việt, xây dựng mô hình Neural.

Kết quả của mô hình có thể dự đoán được các đoạn văn bản và các bài báo điện tử với độ chính xác khá ổn.

Trong xử lý văn bản đầu vào phần quan trọng nhất là phân tách từ tiếng Việt, do tiếng Việt sẽ gồm các từ hoặc các liên từ mới hình thành một tiếng có nghĩa, vì vậy việc xử lý phân tách từ này khá khó khăn. Em đã xây dựng hàm tách từ bằng phương pháp cổ điển là sử dụng bộ từ điển tiếng Việt, việc tiếp cận phương pháp này khá dễ hiểu. Tuy nhiên nó cũng có nhược điểm là bộ từ điển của em chưa đầy đủ và có những từ không hoàn toàn chính xác dẫn đến việc không tách được những từ mới hoặc những từ không có trong từ điển. Vì vậy phải đòi hỏi bộ từ điển với số lượng từ vựng rất lớn.

Về phần đánh giá hiệu quả của mô hình thì mô hình đã cho ra các dự đoán có độ chính xác khá cao

Em cũng đã thiết kế được giao diện cho việc phân loại văn bản theo thời gian thực, tuy giao diện không có nhiều chức năng nhưng cũng thực hiện được chức năng chính là dự đoán các đoạn văn bản hoặc bài báo điện tử.

5.2. Hướng phát triển

Do mới tiếp cận lĩnh vực xử lý ngôn ngữ nên việc tối ưu hoá các bước xử lý và xây dựng mô hình còn chưa hợp lý. Để có thể tiếp tục phát triển hơn cho mô hình, em có thể thực hiện các hướng sau:

Thay đổi cách tiếp cận tách từ tiếng Việt và vector hoá văn bản theo phương pháp tối ưu hơn cách cổ điển là sử dụng tập từ điển có sẵn, xây dựng thêm các mô hình phân loại với các thuật toán khác như SVM, Decision Tree, KNN.

Về phân giao diện dự đoán thời gian thực có thể thêm database chứa các bài báo điện tử mà ta đưa vào dự đoán để việc phân loại văn bản thực tế hơn.

TÀI LIỆU THAM KHẢO

- [1]. Chris Albon (2018). *Machine Learning with Python Cookbook*, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, United States of America
- [2]. Lưu Tuấn Anh. *Natural Language Processing*. <<http://viet.jnlp.org/>>, xem ngày 17/6/2022
- [3]. Nguyễn Trần Thiên Thanh, Trần Khải Hoàng (2005). *Tìm hiểu các phương pháp tiếp cận bài toán phân loại văn bản và xây dựng phân mềm phân loại tin tức báo điện tử*, Luận văn Cử nhân tin học, Trường Đại học Khoa học Tự nhiên.
- [4]. Lê Vĩnh Phú, Diệp Minh Hoàng (2014). *Phân loại tin tức tiếng việt sử dụng các phương pháp học máy*, Luận văn tốt nghiệp Đại học, Trường Đại học Bách khoa.
- [5]. Thuận Bùi Anh (2021). *[Machine learning] Làm thế nào để đánh giá một mô hình Máy học?*, <<http://tutorials.aiclub.cs.uit.edu.vn/index.php/2021/05/18/evaluation/>>, xem ngày 17/6/2022
- [6]. Vũ Hữu Tiệp (2016 - 2020). *Machine learning cơ bản*

Nguồn dữ liệu:

- [1]. Cong Duy Vu Hoang (2019). <<https://github.com/duyvuleo/VNTC>>, xem ngày 5/6/2022.
- [2]. Van Duyet Le (2015). < <https://github.com/stopwords/vietnamese-stopwords> >, xem ngày 7/6/2022.
- [3]. Vũ Anh (2018), <<https://github.com/undertheseanlp/dictionary>>, xem ngày 16/6/2022.