

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



**PBL6: DỰ ÁN CHUYÊN NGÀNH
KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO**

**TÊN ĐỀ TÀI: *Hệ thống phân biệt tin giả trong
lĩnh vực chính trị***

Giảng viên hướng dẫn :	TS. Ninh Khánh Duy
Doanh nghiệp hướng dẫn :	KS. Hoàng Thế Kiệt (EM & AI)
Sinh viên thực hiện:	
Hồ Tài	LỚP: 19TCLC_DT1
Trần Long Duy	LỚP: 19TCLC_DT2
Nguyễn Gia Khánh	LỚP: 19TCLC_DT3
Trương Công Văn	LỚP: 19TCLC_DT3

Đà Nẵng, 12/2022

MỤC LỤC

MỤC LỤC	i
DANH MỤC HÌNH VẼ	iii
MỞ ĐẦU	vii
1. TỔNG QUAN ĐỀ TÀI	1
1.1. Lý do chọn đề tài	1
1.2. Giới thiệu bài toán	1
2. CƠ SỞ LÝ THUYẾT	1
2.1. Ý tưởng.....	1
2.2. Cơ sở lý thuyết.....	2
• Bag of Words (BoW).....	3
• TF-IDF.....	3
• Distributional Embedding.....	3
3. THUẬT TOÁN TRIỂN KHAI.....	3
3.1. Phát biểu bài toán.....	3
3.2. Thuật toán	3
3.2.1. K-nearest Neighbor	3
3.2.2. Naive Bayes.....	5
3.2.3. Support Vector Machine	5
3.2.4. Long Short term Memory.....	9
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ	11
4.1. Tổ chức chương trình.....	11
4.2. Ứng dụng học máy ở trong chương trình	23
4.3. Ứng dụng kiểm thử:.....	29
4.4. Ngôn ngữ cài đặt.....	32
4.5. Kết quả.....	33
4.5.1. Giao diện chính của chương trình	33

4.5.2.	Kết quả thực thi của chương trình	34
4.5.3.	Nhận xét đánh giá.....	34
5.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	35
5.1.	Kết luận.....	35
5.2.	Hướng phát triển	35
	TÀI LIỆU THAM KHẢO	36
	PHỤ LỤC	37
	Tất cả mã nguồn và cài đặt được lưu trữ tại.	37

DANH MỤC HÌNH VẼ

Hình 3.1	Công thức tính khoảng cách
Hình 3.2	Ví dụ minh họa K-nearest Neighbor
Hình 3.3	Siêu mặt phẳng tối ưu và biên
Hình 3.4	Kiến trúc LSTM
Hình 4.3	Số lượng lượng từ các văn bản trong tập dữ liệu
Hình 4.4	50 từ phổ biến khi dữ liệu được tiền xử lý
Hình 4.5	Word Cloud dữ liệu tin giả khi chưa được tiền xử lý
Hình 4.6	Word Cloud dữ liệu tin thật khi chưa được tiền xử lý
Hình 4.7	Số lượng lượng từ các văn bản trong tập dữ liệu sau khi tiền xử lý
Hình 4.8	50 từ phổ biến khi dữ liệu sau khi tiền xử lý
Hình 4.9	Word Cloud dữ liệu tin giả sau khi tiền xử lý
Hình 4.10	Word Cloud dữ liệu tin thật sau khi tiền xử lý
Hình 4.11	Trực quan hoá văn bản sau khi mã hoá bằng TF-IDF
Hình 4.12	Trực quan hoá văn bản sau khi mã hoá bằng Word2Vec
Hình 4.13	Mô tả Vector phân chia 2 lớp
Hình 4.15	Kết quả của độ chính xác và mất mát của mô hình LSTM
Hình 4.15	Ma trận nhầm lẫn khi dự đoán bằng thuật toán KNN
Hình 4.16	Ma trận nhầm lẫn khi dự đoán bằng thuật toán SVM
Hình 4.17	Ma trận nhầm lẫn khi dự đoán bằng thuật toán NB
Hình 4.18	Ma trận nhầm lẫn khi dự đoán bằng thuật toán LSTM
Hình 4.19	Tốc độ của mỗi lần dự đoán của thuật toán
Hình 4.20	Giao diện của chương trình
Hình 4.21	Kết quả thực thi chương trình

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
SVM	Support Vector Machine	Máy vectơ hỗ trợ
RNN	Recurrent neural network	Mạng thần kinh hồi quy
KNN	K-Nearest Neighbors	K láng giềng gần nhất
LSTM	Long Short Term Memory	Bộ nhớ dài-ngắn hạn

LỜI CẢM ƠN

Được sự đồng ý của giảng viên TS Ninh Khánh Duy và cán bộ công ty cổ phần EM & AI Hoàng Thế Kiệt, nhóm của chúng tôi đã thực hiện đồ án: “Hệ thống phân loại tin giả trong lĩnh vực chính trị”.

Xin chân thành cảm ơn giảng viên TS Ninh Khánh Duy và cán bộ công ty cổ phần EM & AI Hoàng Thế Kiệt, người đã trực tiếp hướng dẫn đề tài cho nhóm chúng tôi. Trong quá trình thực hiện đồ án, các thầy đã chỉ bảo và truyền đạt những kiến thức khoa học quý báu, đồng thời cũng đưa những góp ý thiết thực giúp nhóm tôi hoàn thành dự án này.

MỞ ĐẦU

Ngày nay, cùng với sự gia tăng ngày càng nhiều về số lượng thông tin trên không gian internet điều đó đặt ra nhiều vấn đề về độ tin cậy của thông tin. Một vấn đề quan trọng là lượng người dùng internet ngày càng nhiều và đang có xu hướng trẻ hóa dẫn đến hệ quả là những người tham gia vào mạng lưới internet không được trang bị đầy đủ kiến thức để có thể phân biệt được những nguồn tin đáng tin cậy hay là không. Chính trị là một trong những lĩnh vực thông tin mà nếu chúng ta không được trang bị đầy đủ những kiến thức cần thiết thì việc bị những thông tin sai sự thật ảnh hưởng đến suy nghĩ cũng như hành động sẽ gây ra những hậu quả nghiêm trọng cho chính mình và xã hội. Để có thể hạn chế những hậu quả không đáng có như trên thì việc nhận dạng được tin giả một cách tự động là rất cần thiết.

Thông tin trên internet được chia làm ba loại chính là văn bản, âm thanh và hình ảnh. Mà ngôn ngữ tự nhiên của con người tồn tại dưới dạng tiếng nói hoặc văn bản nên việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người được gọi là xử lý ngôn ngữ tự nhiên (NLP). Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, 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). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. 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, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động. Đối với việc phân loại tin giả trong lĩnh vực chính trị thì việc kết hợp của lĩnh vực khai phá dữ liệu, xử lý dữ liệu và học máy sẽ giúp việc nhận dạng tin giả ở lĩnh vực chính trị trở nên dễ dàng hơn. Tính đến nay, việc xử lý bài toán phân loại tin giả trong lĩnh vực chính trị có nhiều hướng tiếp cận khác nhau, tuy nhiên các tài liệu liên quan đa phần áp dụng cho tiếng

Anh. Nên việc xây dựng hệ thống phân loại tin giả trong lĩnh vực chính trị dùng cho tiếng Việt còn nhiều hạn chế.

Mục đích của đề án phân loại tin giả trong lĩnh vực chính trị là tìm hiểu tổng quan về bài toán phân loại tin giả, tập trung và phân tích các phương pháp tiếp cận, các phương thức hoạt động và xử lý dữ liệu chung của bài toán này. Đánh giá những thành phần và những yếu tố quan trọng trong việc khai phá ngữ nghĩa. Tìm hiểu thuật toán học máy, tập trung vào phương thức và trích chọn đặc trưng. Dự án tiếp cận nhiều phương pháp học máy như K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Long Short term memory. Ở mỗi phương pháp tiếp cận, nhóm lựa chọn nhiều phương pháp làm giàu tập dữ liệu, cải thiện cách trích chọn đặc trưng để thông qua đó mà tăng độ chính xác mô hình.

1. TỔNG QUAN ĐỀ TÀI

1.1. Lý do chọn đề tài

Hiện nay, cùng với sự phát triển chung của khoa học kỹ thuật và công nghệ thông tin đã đem đến cho con người khả năng tiếp cận với tri thức khoa học một cách nhanh chóng, cụ thể như: thư viện điện tử, cổng thông tin điện tử, báo mạng, các ứng dụng tìm kiếm, ..., đã giúp con người thuận tiện hơn trong việc trao đổi, cập nhật thông tin trên toàn cầu thông qua mạng Internet.

Tuy nhiên thời gian qua, các thế lực thù địch, phản tử cơ hội, phản động triệt để lợi dụng các phương tiện truyền thông, mạng xã hội để thực hiện nhiều phương thức, thủ đoạn tiến công nhằm tác động vào tư tưởng, tâm lý, tình cảm của đối tượng tiếp nhận, thông qua việc đưa ra các thông tin giả, tin đồn thất thiệt chống phá đại hội Đảng bộ các cấp. Trên internet, mạng xã hội và một số “diễn đàn”, các thế lực thù địch, phản tử cơ hội cố tình đưa nhiều thông tin sai lệch, xuyên tạc.

Chính vì vậy, nhóm chúng tôi thực hiện đề tài ‘**Phân loại tin giả trong lĩnh vực chính trị**’ để phòng, chống thông tin giả, tin đồn thất thiệt.

1.2. Giới thiệu bài toán

1.2.1. Mục tiêu

Xây dựng hệ thống phân loại tin giả trong lĩnh vực chính trị.

- Đầu vào là nội dung của bài báo thuộc lĩnh vực chính trị.
- Đầu ra trả về là tin thật hoặc tin giả có độ tin cậy cao.

1.2.2. Giải pháp

Nghiên cứu, ứng dụng các mô hình học máy và học sâu vào quy trình xử lý ngôn ngữ tự nhiên. Từ đó có thể huấn luyện được các mô hình có độ tin cậy cao áp dụng vào hệ thống phân loại lĩnh vực trong lĩnh vực chính trị.

2. CƠ SỞ LÝ THUYẾT

2.1. Ý tưởng

Ý tưởng dựa trên quy trình xử lý ngôn ngữ tự nhiên gồm 6 bước:

- Thu thập dữ liệu
- Tiền xử lý văn bản
- Khai phá dữ liệu
- Mã hóa văn bản.

- Huấn luyện mô hình
- Triển khai

2.2. Cơ sở lý thuyết

Quy trình xử lý ngôn ngữ tự nhiên

2.2.1. Thu thập dữ liệu.

Là quá trình thu thập các dữ liệu cần thiết dùng để thực hiện dự án. Việc thu thập dữ liệu phục vụ cho xử lý ngôn ngữ tự nhiên có thể được thực hiện trên nhiều hình thức khác nhau. Phổ biến trong đó là thu thập dữ liệu từ các website.

2.2.2. Tiền xử lý văn bản.

Tiền xử lý dữ liệu là một kỹ thuật khai thác dữ liệu bao gồm chuyển đổi dữ liệu bao gồm chuyển đổi dữ liệu thô thành định dạng dễ hiểu. Dữ liệu trong thế giới thực thường không đầy đủ, không nhất quán hoặc thiếu một số hành vi hoặc xu hướng nhất định và có khả năng chứa nhiều lỗi. Tiền xử lý dữ liệu là một phương pháp đã được chứng minh để giải quyết các vấn đề đó.

2.2.3. Khai phá dữ liệu.

Khai thác dữ liệu đảm bảo một công ty đang thu thập và phân tích dữ liệu đáng tin cậy. Nó thường là một quá trình có cấu trúc cứng nhắc hơn, chính thức xác định một vấn đề, tập hợp dữ liệu liên quan đến vấn đề và cố gắng xây dựng một giải pháp. Do đó, khai thác dữ liệu giúp một doanh nghiệp trở nên có lợi nhuận cao hơn, hiệu quả hơn hoặc hoạt động mạnh hơn. Khai thác dữ liệu có thể trông rất khác nhau giữa các ứng dụng, những quy trình tổng thể có thể được sử dụng với hầu hết mọi ứng dụng mới hoặc cũ. Về cơ bản, bất kỳ loại dữ liệu nào cũng có thể được thu thập và phân tích, và hầu hết mọi vấn đề kinh doanh đều dựa trên bằng chứng có thể giải quyết được bằng cách sử dụng khai thác dữ liệu. Mục tiêu cuối cùng của khai thác dữ liệu là lấy các thông tin thô và xác định xem có sự gắn kết hay tương quan giữa các dữ liệu hay không. Lợi ích này của việc khai thác dữ liệu cho phép một công ty tạo ra giá trị với thông tin họ có trong tay mà nếu không sẽ không quá rõ ràng. Mặc dù các mô hình dữ liệu có thể phức tạp, nhưng chúng cũng có thể mang lại kết quả hấp dẫn, các xu hướng ẩn giấu và đề xuất các chiến lược độc đáo.

2.2.4. Mã hóa văn bản.

Thông thường, máy tính không thể hiểu được ý nghĩa các từ. Như vậy, để xử lý được ngôn ngữ tự nhiên, ta cần có một phương pháp để biểu diễn văn bản dưới dạng

mà máy tính có thể hiểu được. Phương pháp tiêu chuẩn để biểu diễn văn bản đó là biểu diễn các văn bản theo vector. Trong đó, các từ/cụm từ thuộc kho tài liệu ngôn ngữ được ánh xạ thành những vector trên hệ không gian số thực. Các phương pháp mã hóa văn bản có thể kể đến như:

- Bag of Words (BoW)
- TF-IDF
- Distributional Embedding
- Word2Vec
- Glove
- FastTText

2.2.5. Huấn luyện mô hình.

Từ dữ liệu đã được mã hóa chúng ta tiến hành lựa chọn thuật toán huấn luyện mô hình học máy. Lựa chọn thuật toán học máy phù hợp với bài toán cần giải quyết, có thể kết hợp nhiều thuật toán và phương pháp tạo ra mô hình thích hợp.

Cần chia dữ liệu thành 2 phần : phần để huấn luyện (training data) và phần để kiểm thử (testing data), tùy theo mô hình để chia tỉ lệ thích hợp. Tiến hành thử nghiệm và đánh giá mô hình để có sự điều chỉnh phù hợp.

2.2.6. Triển khai.

Sau khi huấn luyện mô hình với kết quả mong đợi thì đây là bước triển khai mô hình vào thực tiễn.

3. THUẬT TOÁN TRIỂN KHAI

3.1. Phát biểu bài toán

Đầu vào: Văn bản (bản tin) thuộc lĩnh vực chính trị.

Đầu ra: Dự đoán của mô hình (tin giả hoặc tin thật)

3.2. Thuật toán

Các thuật toán được sử dụng để huấn luyện mô hình gồm K-nearest Neighbor, Naive Bayes, Support Vector Machine, Long Short Term Memory.

3.2.1. K-nearest Neighbor

K-Nearest Neighbor (KNN) là phương pháp phân lớp các đối tượng dựa vào khoảng cách gần nhất. Mỗi đối tượng được xác định bằng K láng giềng của nó bằng cách đo khoảng cách giữa chúng và các điểm láng giềng dưới độ đo Euclid, Manhattan hoặc Minkowski.

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Hình 3.1 Công thức tính khoảng cách



Hình 3.2 Ví dụ minh họa K-nearest Neighbor

Các bước thực thực hiện thuật toán:

1. Xác định giá trị tham số K (số láng giềng gần nhất)
2. Tính khoảng cách giữa đối tượng cần phân lớp (Query Point) với tất cả các đối tượng trong training data (thường sử dụng khoảng cách Euclid)
3. Sắp xếp khoảng cách theo thứ tự tăng dần và xác định K láng giềng gần nhất với điểm cần phân lớp.

4. Lấy tất cả các lớp của K láng giềng gần nhất đã xác định
5. Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho điểm cần phân lớp.

3.2.2. Naive Bayes

Thuật toán Naive Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại. Chủ yếu được sử dụng trong phân loại văn bản bao gồm tập dữ liệu huấn luyện chiều cao. Naive Bayes Classification là một trong những bài toán phân loại đơn giản và hiệu quả nhất giúp xây dựng mô hình học máy có thể đưa ra dự đoán nhanh.

Định lý Bayes:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Với:

$P(A|B)$ là xác suất hậu định (Posterior probability) hay là xác suất của giả thuyết A đối với sự kiện B được quan sát.

$P(B|A)$ là xác suất khả dĩ (likelihood probability) hay là xác suất của bằng chứng cho thấy xác suất của một giả thuyết là đúng.

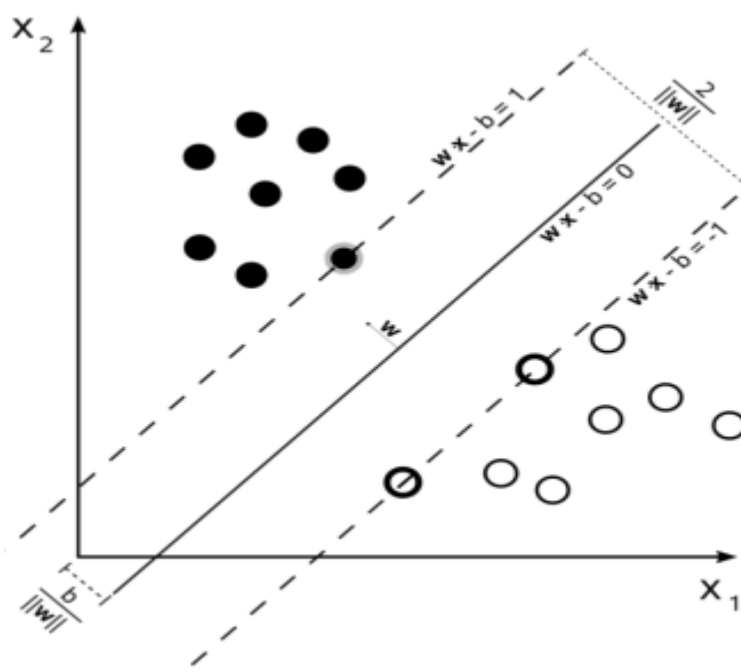
$P(A)$ là xác suất tiên định (prior probability) hay là xác suất của giả thuyết trước khi quan sát bằng chứng.

$P(B)$ là xác suất biên hay là xác suất chứng cứ.

Sau khi tính được xác suất hậu định của các giả thuyết (lớp), giả thuyết (lớp) nào có xác suất hậu định lớn nhất thì sự kiện đó sẽ được dự đoán là giả thuyết (lớp) đó.

3.2.3. Support Vector Machine

Support Vector Machine là một thuật toán trong thống kê và khoa học máy tính cho một tập hợp các phương pháp học có giám sát liên quan đến nhau nhằm để phân loại và phân tích hồi quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Thuật toán SVM chia hai lớp dữ liệu bằng một siêu mặt phẳng $d-1$ chiều khi số chiều của dữ liệu huấn luyện là d . Hình xxx là ví dụ phân tách dữ liệu thuộc hai lớp sử dụng SVM. Trong đó, $w \cdot x - b = 0$ là siêu mặt phẳng thể hiện sự phân tách dữ liệu.



Hình 3.3 Siêu mặt phẳng tối ưu và biên

Cho trước tập huấn luyện được biểu diễn trong không gian véc tơ, trong đó mỗi văn bản là một điểm, phương pháp SVM dạng chuẩn tìm ra một siêu mặt phẳng 3D quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng với lớp (+) và lớp (-). Hiệu quả xác định siêu mặt phẳng này được quyết định bởi khoảng cách của điểm gần mặt phẳng nhất của mỗi lớp. Khoảng cách càng lớn thì mặt phẳng quyết định càng tốt đồng nghĩa với việc phân loại càng chính xác và ngược lại. Mục đích cuối cùng của phương pháp là tìm khoảng cách biên lớn nhất.

- **Mục đích:** Dùng để phân lớp dữ liệu mới thuộc lớp nào.
- **Dữ liệu vào:** Cho trước một tập dữ liệu huấn luyện đã gán nhãn thuộc lớp -1 hoặc +1.
- **Dữ liệu ra:** Dùng để tìm ra nhãn cho các dữ liệu mới.
- **Mô tả thuật toán:** Ta có một tập huấn luyện D , một tập gồm n điểm có dạng.

$$D = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

Với y_i mang giá trị +1 hoặc -1, ngầm định lớp chứa điểm x_i . Mỗi x_i là một véc tơ thực p - chiều. Ta cần tìm siêu mặt phẳng có lề lớn nhất chia tách các điểm x_i thuộc lĩnh vực quan tâm và được gán nhãn $y_i = 1$ và các điểm x_i thuộc lĩnh vực không quan

tâm và được gán nhãn $y_i = -1$. Mỗi siêu mặt phẳng đều có thể được viết như là một tập các điểm x thỏa mãn phương trình:

$$w \cdot x - b = 0$$

Với (\cdot) kí hiệu cho tích vô hướng. Véc tơ trọng số w , nó vuông góc với siêu mặt phẳng. Tham số $\frac{b}{\|w\|}$ xác định độ lệch của siêu mặt phẳng từ nó đến vector w .

Phương trình trên tương đương với phương trình sau:

$$C + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = 0 \quad (1.4)$$

Tương đương với công thức :

$$C + \sum_{i=1, \dots, n} w_i \cdot x_i = 0 \quad (1.5)$$

Với $w = w_1 + w_2 + \dots + w_n$ là bộ hệ số siêu mặt phẳng hay là vector trọng số, C là độ dịch, khi thay đổi w và C thì hướng và khoảng cách từ gốc tọa độ đến siêu mặt phẳng thay đổi.

Chúng ta cần chọn w và b để cực đại lề, hay khoảng cách giữa hai siêu mặt phẳng song song sao cho chúng càng xa càng tốt trong khi vẫn phân chia tốt dữ liệu. Các siêu mặt phẳng ấy được xác định bằng:

$$w \cdot x - b = 1$$

và

$$w \cdot x - b = -1$$

Để ý rằng nếu một dữ liệu huấn luyện có thể được chia tách một cách tuyến tính (bằng một đường thẳng), chúng ta có thể chọn hai siêu mặt phẳng của lề sao cho không có điểm nào ở giữa chúng và cố gắng cực đại khoảng cách giữa chúng. Bằng hình học, chúng ta tìm được khoảng cách giữa hai siêu mặt phẳng là $\frac{2}{\|w\|}$ và ta muốn cực tiểu giá trị $\|w\|$. Để tránh các điểm dữ liệu rơi vào bên trong lề, chúng ta thêm vào các điều kiện sau, với mỗi i ta có:

$$w \cdot x_i - b \geq 1 \quad \text{nếu } y_i = 1 \text{ (đối với lớp thứ nhất)}$$

hoặc

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \text{ nếu } y_i = -1 \text{ (đối với lớp thứ hai)}$$

Có thể viết gọn lại như sau :

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \text{ với mọi } 1 \leq i \leq n$$

Ta có thể gom chúng lại trong một bài toán tối ưu:

Cực tiểu (theo \mathbf{w} , b): $\|\mathbf{w}\|$ với điều kiện (với mọi $i = 1, \dots, n$)

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

Các vấn đề tối ưu hóa được trình bày trong phần trước là khó khăn để giải quyết bởi vì nó phụ thuộc vào $\|\mathbf{w}\|$, chỉ tiêu \mathbf{w} , trong đó bao gồm một căn bậc hai. May mắn thay, nó có thể làm thay đổi phương trình bằng cách thay thế $\|\mathbf{w}\|$ với $\frac{1}{2} \|\mathbf{w}\|^2$

(Yếu tố $1/2$ đang được sử dụng để thuận tiện trong toán học) mà không thay đổi các giải pháp (tối thiểu của bản gốc và phương trình sửa đổi có \mathbf{w} và b). Đây là vấn đề tối ưu hóa một phương trình bậc hai. Rõ ràng hơn:

Cực tiểu (trong \mathbf{w} , b) : $\frac{1}{2} \|\mathbf{w}\|^2$ (với bất kỳ $i = 1, \dots, n$)

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

Đưa về phương trình Lagrange với số nhân α_i [7]:

$$\min_{\mathbf{w}, b, \alpha} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\} \quad (1.6)$$

Máy véc tơ hỗ trợ SVM là quá trình tìm ra các siêu mặt phẳng phụ thuộc vào tham số vectơ trọng số \mathbf{w} và độ dịch b . Mục tiêu của phương pháp SVM là ước lượng \mathbf{w} và b để cực đại hóa lề giữa các lớp dữ liệu dương và âm. Các giá trị khác nhau của lề cho ta các họ siêu mặt phẳng khác nhau và lề càng lớn thì năng lực của máy học càng giảm. Như vậy, cực đại hóa lề thực chất là việc tìm một máy học có năng lực nhỏ nhất. Quá trình phân loại là tối ưu khi sai số phân loại là cực tiểu. Sau khi đã tìm được phương trình của siêu mặt phẳng bằng thuật toán SVM, áp dụng công thức này để tìm ra nhãn lớp cho các dữ liệu mới.

Nếu dữ liệu học không tách rời tuyến tính, thêm biến η_i và thay phương trình trên bằng phương trình:

$$\min_{\mathbf{W}, b, \eta} C = \sum_{i=1}^n \eta_i + \frac{1}{2} \|\mathbf{W}\|^2$$

$$\text{Với } y_i[\mathbf{W} \cdot \mathbf{x}_i - b] + \eta_i \geq 1$$

$$\eta_i \geq 0, \quad i = 1, \dots, n$$

(1.7)

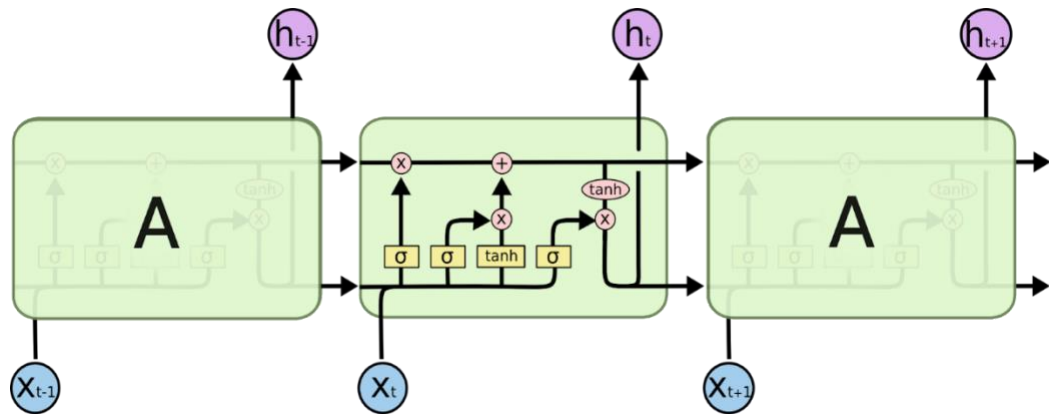
Từ đó ta có phương trình tổng quát của siêu mặt phẳng tìm ra được bởi thuật toán SVM là:

$$\hat{f}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) = C + \sum w_i \cdot \mathbf{x}_i \quad (1.8)$$

Với $i = 1, \dots, n$. Trong đó n là số dữ liệu huấn luyện.

3.2.4. Long Short term Memory

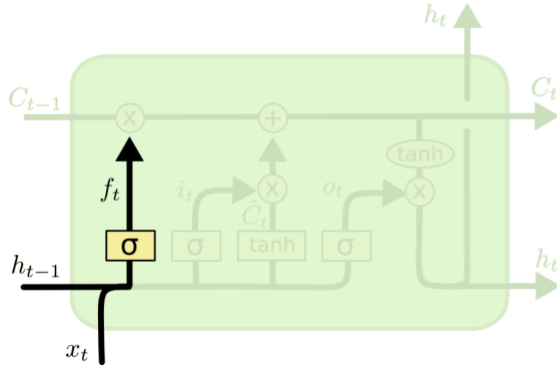
Long short-term memory(LSTM) là một mạng cải tiến của RNN, nó có khả năng học được các phụ thuộc xa (long-term dependency). LSTM được thiết kế để tránh vấn đề phụ thuộc xa. Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ không cần phải huấn luyện để có thể nhớ được.



Hình 3.4. Kiến trúc LSTM

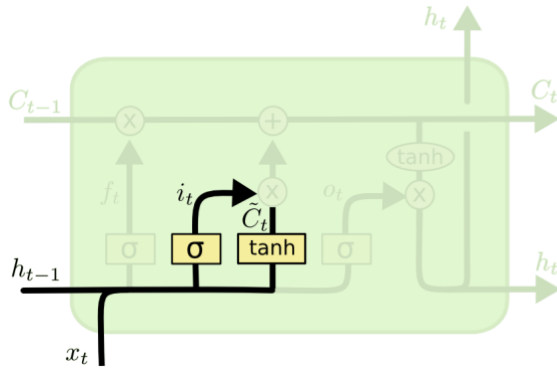
Bước đầu tiên trong LSTM là quyết định những thông tin nào cần được loại bỏ từ cell internal state ở bước trước đó c_{t-1} . Activation value f_t của forget gate tại bước thời gian t được tính dựa trên giá trị đầu vào hiện tại x_t , giá trị đầu ra h_{t-1} từ tế bào LSTM ở

bước trước đó và bias b_f của forget gate.



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

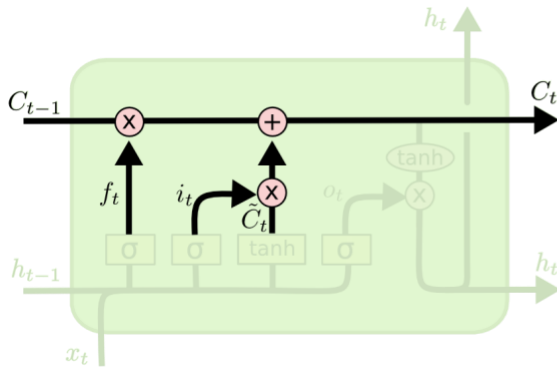
Bước thứ hai, LSTM quyết định những thông tin nào cần được thêm vào cell internal state c_t . Bước này có 2 bước tính toán, đầu tiên là input gate layer (kí hiệu: i_t) quyết định giá trị nào được cập nhập. Tiếp theo là candidate value (kí hiệu: \tilde{c}_t) biểu diễn những thông tin tiềm năng cần thêm vào c_t .



$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

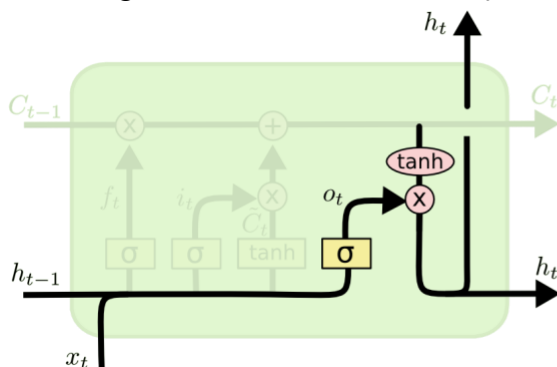
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Bước thứ ba, cập nhập lại giá trị mới của cell internal state c_t .



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Bước cuối cùng, tính giá trị đầu ra h_t . Đầu tiên, dùng lớp sigmoid để quyết định phần nào sẽ được xuất ra. Sau đó, giới hạn giá trị của cell internal state c_t trong khoảng -1 đến 1 bằng hàm tanh và nhân nó với o_t để xuất ra các phần đã được chọn.



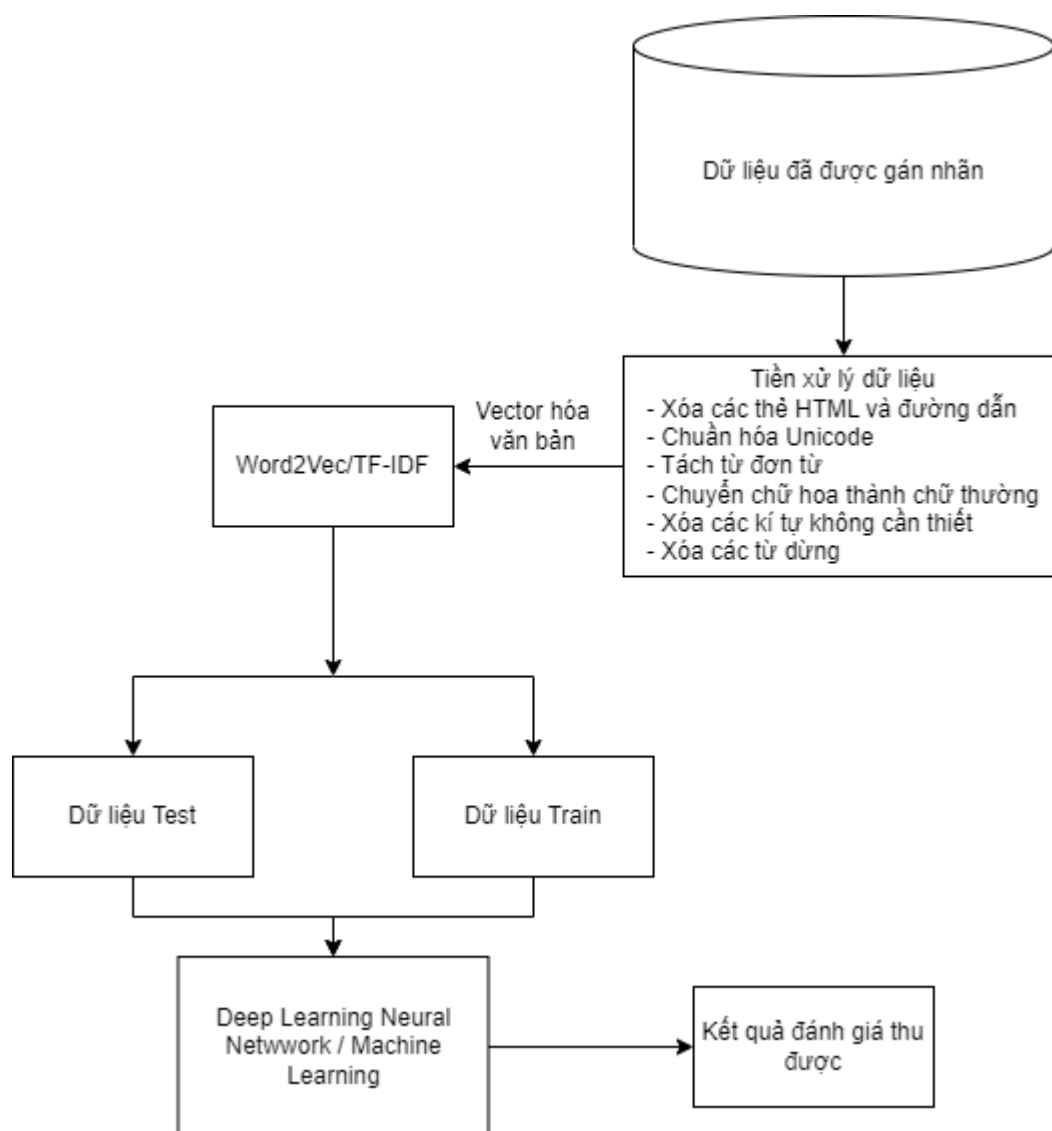
$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1. Tổ chức chương trình

Mô hình tổng quát huấn luyện.



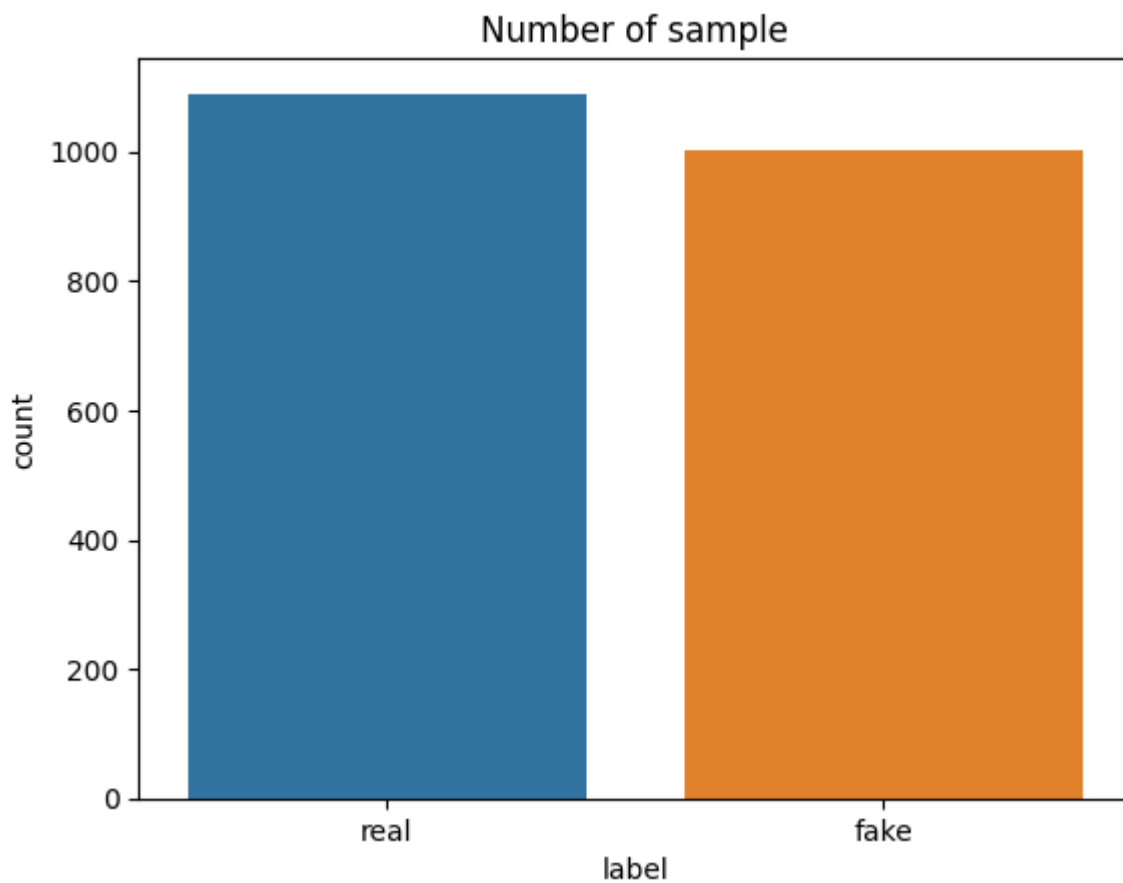
Hình 4.1. Mô hình tổng quát quá trình huấn luyện thuật toán

4.1.1. Thu thập dữ liệu

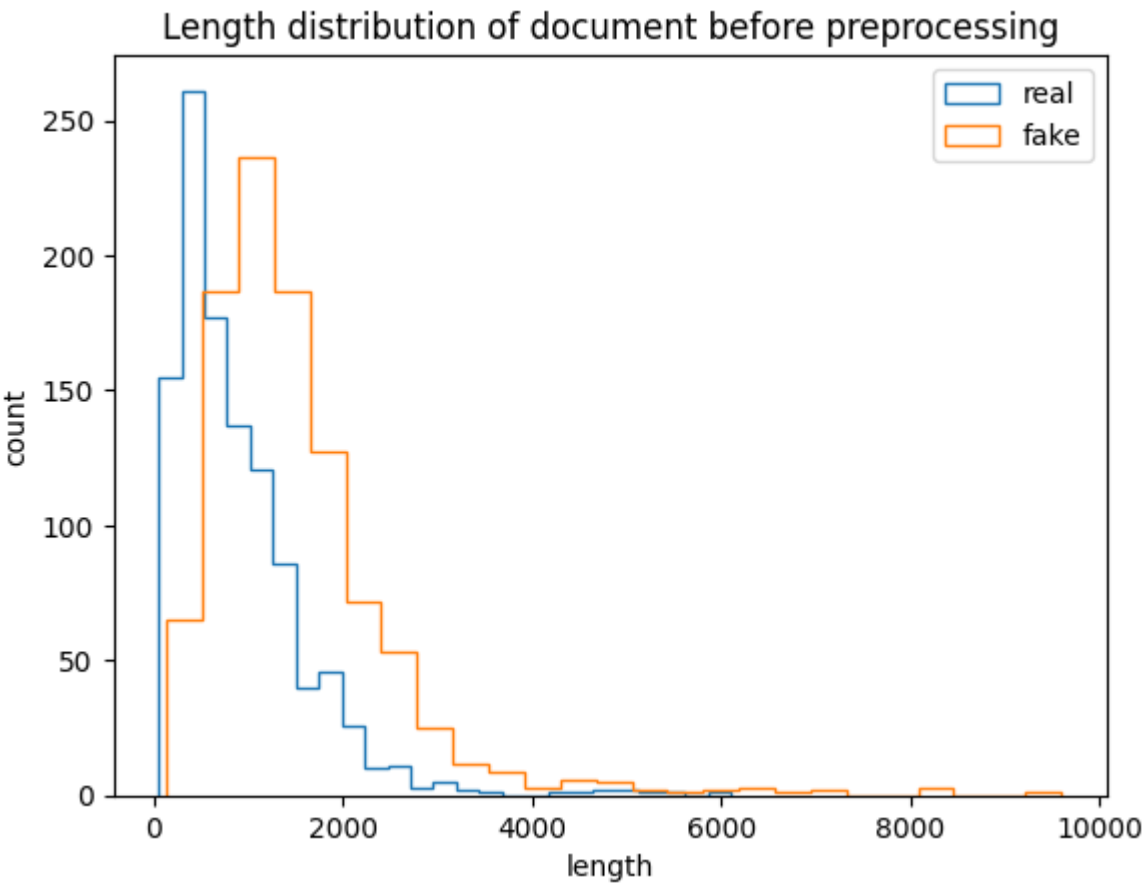
Dữ liệu được thu thập từ nhiều nhiều nguồn và được chia làm hai nhóm chính

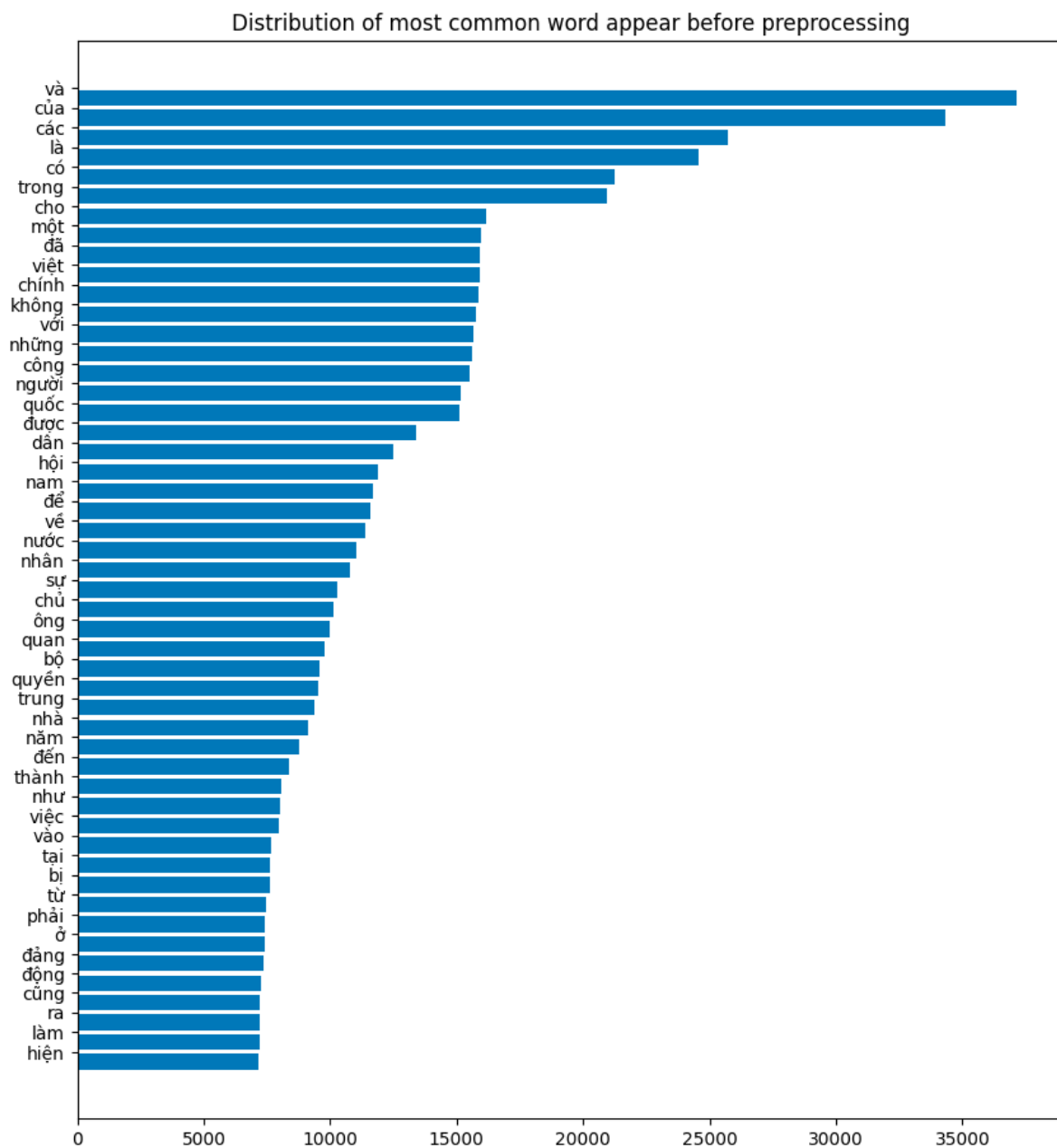
- Nhóm tin thật: <https://vietnamnet.vn> (457 mẫu), <https://thanhnien.vn> (473 mẫu), <https://nhandan.vn> (212 mẫu), <https://dantri.com.vn> (799 mẫu)
- Nhóm tin giả: <https://viettan.org> (1120 mẫu), <http://nhanvanviet.com> (1214 mẫu), <https://quanlambao.blogspot.com> (1117 mẫu)

Từ những dữ liệu trên nhóm đã đánh nhãn thủ công được 1096 mẫu tin thật và 911 mẫu tin giả.



Hình 4.2. Biểu đồ của dữ liệu





Hình 4.3. Số lượng từ các văn bản trong tập dữ liệu.

Hình 4.4. 50 từ phổ biến khi dữ liệu chưa được tiền xử lý.



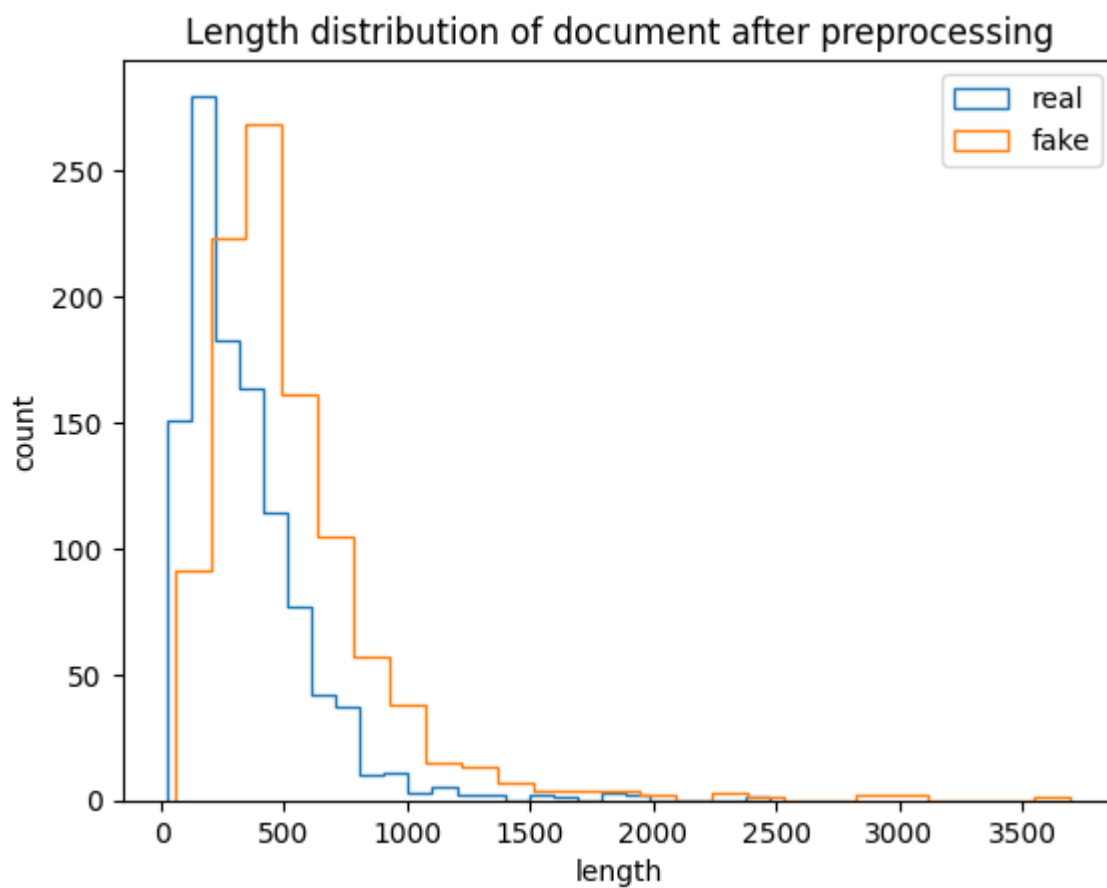
Hình 4.5. Word Cloud dữ liệu tin giả khi chưa được tiền xử lý.

1000

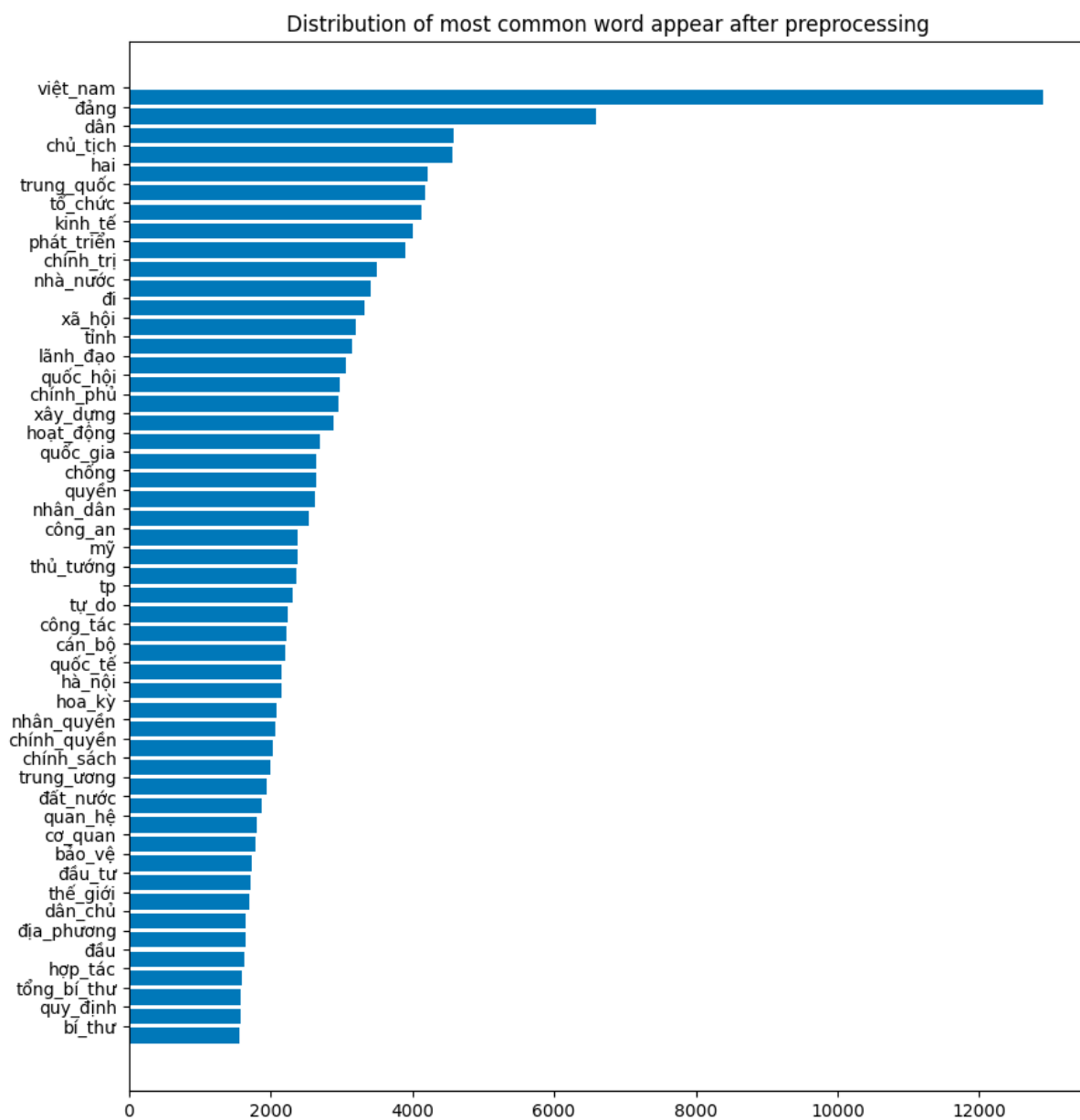
9. *Journal of the American Medical Association*, 2000; 283: 2689-2696.

- Xóa các từ dừng

Tiền xử lý dữ liệu là một bước rất quan trọng, nó sẽ ảnh hưởng đến độ chính xác sau khi huấn luyện mô hình.



Hình 4.7. Số lượng lượng từ các văn bản trong tập dữ liệu sau khi tiền xử lý.



Hình 4.8. 50 từ phổ biến khi dữ liệu sau khi tiền xử lý.



Hình 4.9. Word Cloud dữ liệu tin giả sau khi tiền xử lý.

quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản. Trọng số của mỗi từ được tính theo công thức:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Trong đó:

- $tf_{i,j}$ là số lần từ i xuất hiện trong văn bản j
- df_i là số lượng văn bản chứa từ i
- N là số lượng văn bản trong tập dữ liệu

4.1.3.2 Word2Vec

Word2vec là một kỹ thuật xử lý ngôn ngữ tự nhiên. Thuật toán Word2vec sử dụng một mô hình mạng thần kinh để học các liên kết từ (sự liên quan của từ) từ một kho ngữ liệu văn bản có dung lượng lớn. Sau khi được huấn luyện, mô hình có thể phát hiện các từ đồng nghĩa hoặc gợi ý các từ bổ sung cho một phần của câu. Với cái tên nói lên tất cả, word2vec thể hiện cho mỗi từ riêng biệt với một danh sách cụ thể của các số được gọi là vector. Các vector được lựa chọn cẩn thận sao cho một hàm toán học đơn giản sẽ (độ tương tự cosin giữa các vector) cho biết mức độ của độ tương tự ngữ nghĩa giữa các từ được biểu diễn bằng các vector đó. Word2vec là một nhóm các mô hình có quan hệ với nhau được dùng để sản sinh các nhúng từ (word embedding). Các mô hình này là các mạng thần kinh nông hai lớp, được huấn luyện để tái tạo lại ngữ cảnh ngữ nghĩa của các từ vụng. Word2vec có dữ liệu đầu vào là một ngữ liệu văn bản lớn và đầu ra là một không gian vector, điển hình vài trăm chiều, với mỗi từ duy nhất trong ngôn ngữ học khối liệu (corpus linguistics) được gán cho một vector tương ứng trong không gian vector. Các vector từ được đặt trong không gian vector sao cho những từ chia sẻ chung ngữ cảnh trong kho ngữ liệu có vị trí gần nhau (tính theo độ tương tự ngữ nghĩa) trong không gian.

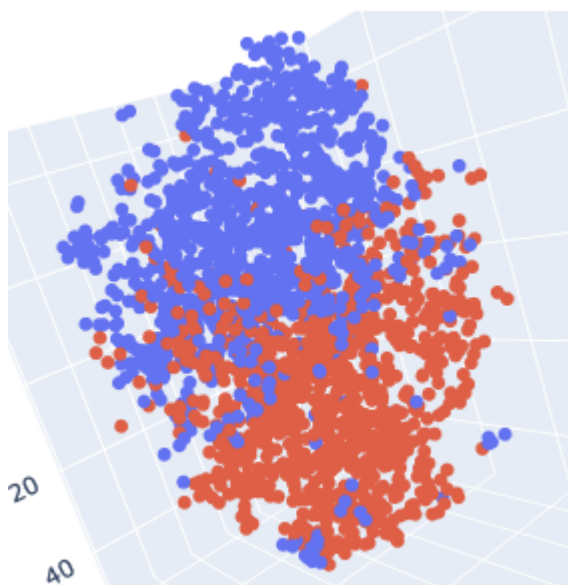
Trong bài toán này quá trình huấn luyện Word2Vec được thực hiện dựa trên các dữ liệu chưa được gán nhãn trong tập dữ liệu. Với số chiều đầu ra của thuật toán là 300 chiều đối với mỗi từ. Để có thể mã hóa một văn bản thì sử dụng phương pháp tính trung bình hệ số Word2Vec của mỗi văn bản. Do đó đầu ra của pha vector hóa của văn bản là một vector 300 chiều.

4.1.4. Trục quan hóa văn bản

Trục quan hóa dữ liệu từ tập dữ liệu huấn luyện và được vector hóa bằng TF-IDF và Word2Vec

4.1.4.1 TF-IDF

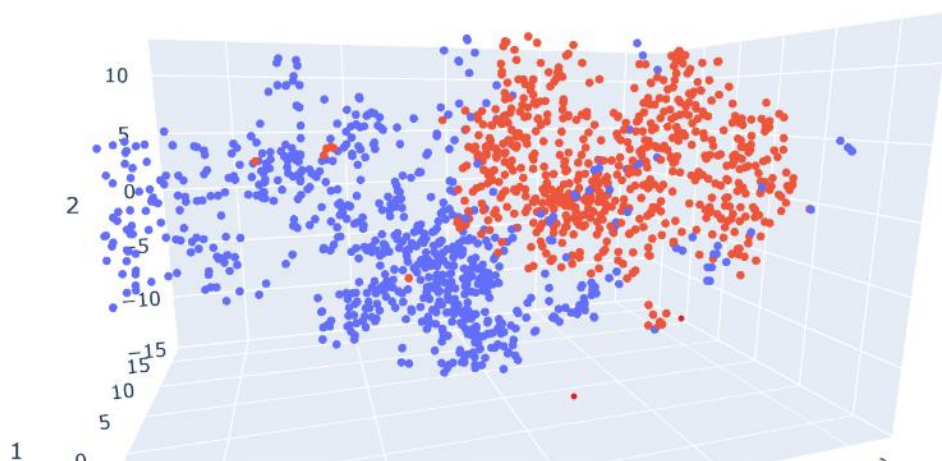
Sử dụng thuật toán giảm chiều t-SNE để trục quan vector 15421 chiều khi mã hóa bằng TF-IDF ở không gian 3 chiều.



Hình 4.11: Trục quan hoá văn bản sau khi mã hoá bằng TF-IDF

4.1.4.2 Word2Vec

Sử dụng thuật toán giảm chiều t-SNE để trục quan vector 300 chiều ở không gian 3 chiều. Mỗi điểm tương ứng với vector trung bình Word2Vec của tất cả các từ trong văn bản. Với màu vàng là những dữ liệu với nhãn là tin thật và màu xanh là nhãn tin giả.



Hình 4.12: Trực quan hoá văn bản sau khi mã hoá word2vec

Qua hai phương pháp mã hóa có thể thấy rằng có sự phân cụm khá rõ ràng giữa dữ liệu tin thật và dữ liệu tin giả.

4.1.5. Kỹ thuật K-Fold Cross Validation:

Cross validation là một phương pháp thống kê được sử dụng để ước lượng hiệu quả của các mô hình học máy. Nó thường được sử dụng để so sánh và chọn ra mô hình tốt nhất cho một bài toán. Kỹ thuật này dễ hiểu, dễ thực hiện và cho ra các ước lượng tin cậy hơn so với các phương pháp khác. Trong bài toán này, chúng tôi áp dụng kỹ thuật này để tìm ra các tham số phù hợp với mô hình, còn được biết đến với bước Parameter Tuning. Sẽ được trình bày kỹ hơn ở phần 4.2.

4.1.6. Xây dựng mô hình học máy và đánh giá mô hình

Dự án xây dựng tổng cộng 4 mô hình, ở mỗi mô hình thông qua kết quả kiểm thử sẽ phân tích những ưu nhược điểm của từng mô hình, đưa ra nguyên nhân hướng giải quyết để cải thiện hiệu quả. Phần này sẽ trình bày chi tiết ở phần 4.2.

4.2. Ứng dụng học máy ở trong chương trình

4.2.1. Mô hình 1: K-nearest Neighbor

- Mô tả

K-Nearest Neighbor (KNN) là phương pháp phân lớp các đối tượng dựa vào khoảng cách gần nhất. Mỗi đối tượng được xác định bằng K láng giềng của nó bằng

cách đo khoảng cách giữa chúng và các điểm láng giềng dưới độ đo Euclid, Manhattan hoặc Minkowski.

- **Kết quả huấn luyện**

KNN có các tham số ảnh hưởng đến kết quả như sau:

- n_neighbor
- weighs
- metric

Với việc sử dụng GridSearchCV với 10 Fold để tìm ra tham số tối ưu cho mô hình, kết quả thu được như sau:

TF-IDF là vector input:

- n_neighbors : 11
- metric: minkowski
- weights : uniform

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	85.23%	96.31%	90.25%
Hyper parameter	89.55%	94.11%	92.33%

Word2Vec là vector input:

- n neighbors : 5
- metric: manhattan
- weights : distance

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	93.23%	98.49%	96.12%
Hyper parameter	93.40%	99.67%	96.52%

- **Đánh giá**

Với mô hình K-nearest Neighbor phương pháp mã hóa bằng Word2Vec mô hình có tỉ lệ chính xác cao hơn lên đến 4%.

4.2.2. Mô hình 2: Naive Bayes Classification

- **Mô tả**

Thuật toán Naive Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại. Chủ yếu được sử dụng trong phân loại văn bản bao gồm tập dữ liệu huấn luyện chiều cao. Naive Bayes Classification là một trong những bài toán phân loại đơn giản và hiệu quả nhất giúp xây dựng mô hình học máy có thể đưa ra dự đoán nhanh.

- **Kết quả huấn luyện**

TF-IDF là vector input

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	91.12%	97.44%	94.25%

Word2Vec là vector input

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	93.01%	94.34%	93.71%

- **Đánh giá**

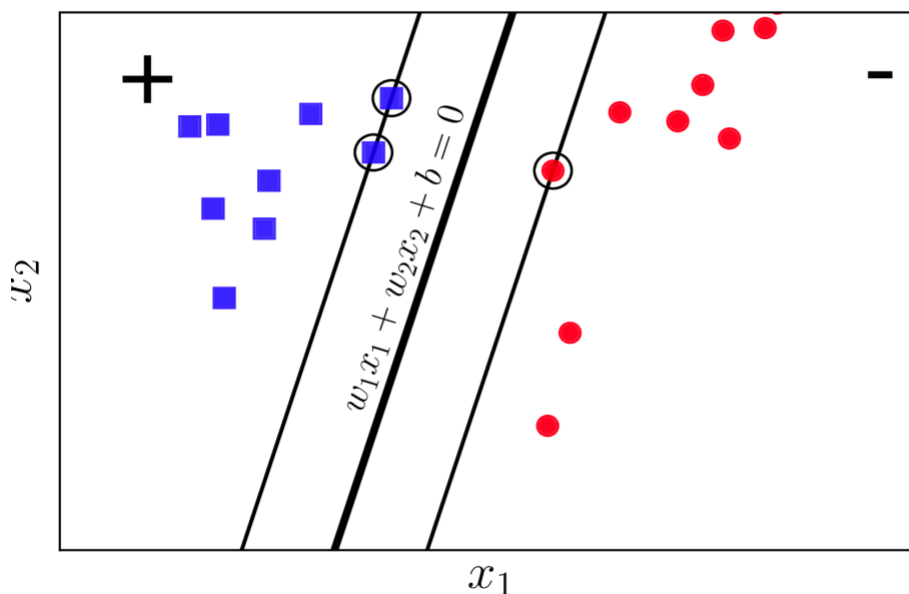
Với mô hình Naive Bayes Classification phương pháp mã hóa bằng TF-IDF mô hình có tỉ lệ chính xác cao hơn 1% so với phương pháp mã hóa bằng Word2Vec.

4.2.3. Mô hình 3: Support Vector Machine

- **Mô tả**

Trong thuật toán này, chúng ta biểu diễn các điểm lên không gian. Sau đó chúng ta thực hiện tìm "đường bay" (*hyper-plane*) phân chia các lớp. Hyper-plane nó

chi hiệu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.



Hình 4.13: Mô tả Vector phân chia 2 lớp

Cho tập dữ liệu học gồm n dữ liệu gán nhãn $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ với $y_i \in \{-1, 1\}$ là một số nguyên xác định lớp của x_i . Mỗi x_i là một văn bản được biểu diễn dưới dạng một vector thực d chiều. Bộ phân lớp tuyến tính (mô hình phân lớp) được xác định thông qua một siêu phẳng có dạng: $\mathbf{f}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - \mathbf{b} = 0$ trong đó: \mathbf{w} là vector pháp tuyến của siêu phẳng và \mathbf{b} đóng vai trò là tham số mô hình. Bộ phân lớp nhị phân được xác định thông qua dấu của $\mathbf{f}(\mathbf{x})$.

Để đảm bảo là luôn tìm được siêu phẳng phân cách có lề cực đại, hàm xác định SVM sẽ làm cực đại hàm sau với vector \mathbf{w} và hằng số \mathbf{b} . Trong đó: \mathbf{w} và hằng số \mathbf{b} xác định mặt phẳng.

1) Correctly classify all training data

$$\left. \begin{aligned} wx_i + b &\geq 1 && \text{if } y_i = +1 \\ wx_i + b &\leq -1 && \text{if } y_i = -1 \\ y_i(wx_i + b) &\geq 1 && \text{for all } i \end{aligned} \right\}$$

2) Maximize the Margin
same as minimize

$$M = \frac{1}{\|\mathbf{w}\|} \\ \frac{1}{2} \mathbf{w}^t \mathbf{w}$$

- Kết quả huấn luyện

SVM có các tham số ảnh hưởng đến kết quả như sau:

- C: chi phí mở rộng margin.
- kernel: một hàm số biến đổi dữ liệu x từ không gian ban đầu thành dữ liệu trong một không gian mới bằng hàm số $\Phi(x)$.
- gamma: xác định việc sử dụng bao nhiêu điểm dữ liệu cho việc xây dựng siêu phẳng phân cách.

Với việc sử dụng GridSearchCV với 10 Fold để tìm ra tham số tối ưu cho mô hình với:

- best_params: {C: 10, gamma=0.1, kernel: "rbf"}

Kết quả thu được như sau:

TF-IDF là vector input

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	97.22%	97.41%	97.32%
Hyper parameter	97.63%	98.2%	97.93%

Word2Vec là vector input

	Fake news accuracy	Real news accuracy	Overall accuracy
Default parameter	97.23%	96.12%	96.62%
Hyper parameter	97.50%	96.60%	97.02%

- Đánh giá

Với mô hình Support Vector Machine phương pháp mã hóa bằng TF-IDF mô hình có tỉ lệ chính xác cao hơn so với mã hóa bằng Word2Vec.

4.2.4. Mô hình 4: Long Short Term Memory

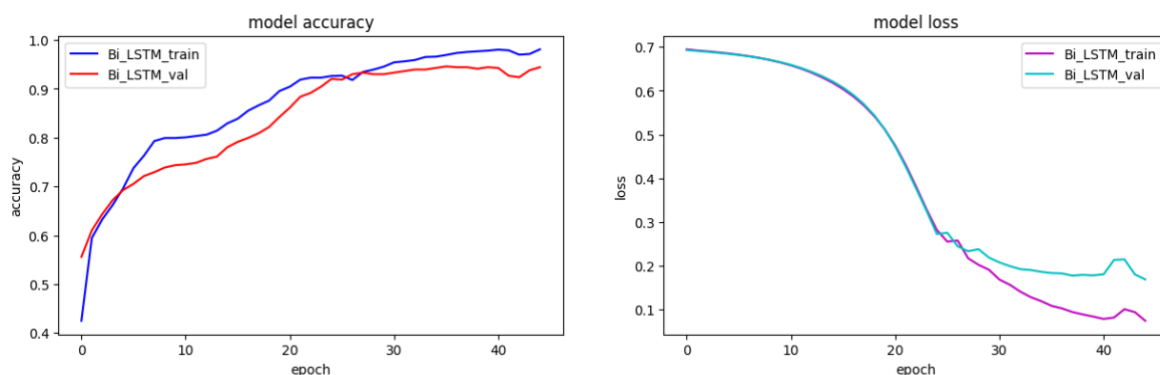
- Mô tả

Long short-term memory(LSTM) là một mạng cải tiến của RNN, nó có khả năng học được các phụ thuộc xa (long-term dependency). LSTM được thiết kế để tránh vấn đề phụ thuộc xa. Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ không cần phải huấn luyện để có thể nhớ được.

Dữ liệu sau khi được xử lý sẽ được tách từ và xây dựng một bộ từ điển bằng Tokenizer. Tiếp theo dữ liệu sẽ được xử lý về độ dài tối đa và đưa vào model để train.

Model sẽ có 5 lớp, lớp đầu vào là lớp embedding với ma trận đầu vào là 600x64, 2 lớp ẩn là lớp Bidirection LSTM có số chiều là 64 và lớp Dense có số chiều là 16 có hàm activation là relu và lớp đầu ra lớp Dense có số chiều là 1 và hàm activation là sigmoid.

- Kết quả



Hình 4.14: Kết quả của độ chính xác và mất mát của mô hình LSTM

Fake news accuracy	Real news accuracy	Overall accuracy
94.32%	95.19%	94.59%

- Đánh giá

Với mô hình Long Short Term Memory độ chính xác của mô hình đạt được là 94.59%.

4.3. Ứng dụng kiểm thử:

4.3.1. Các phương thức đánh giá

- Accuracy:

Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm. Nó có thể được tính toán dễ dàng bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán.

- Recall:

Recall cũng là một metric quan trọng, nó đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive.

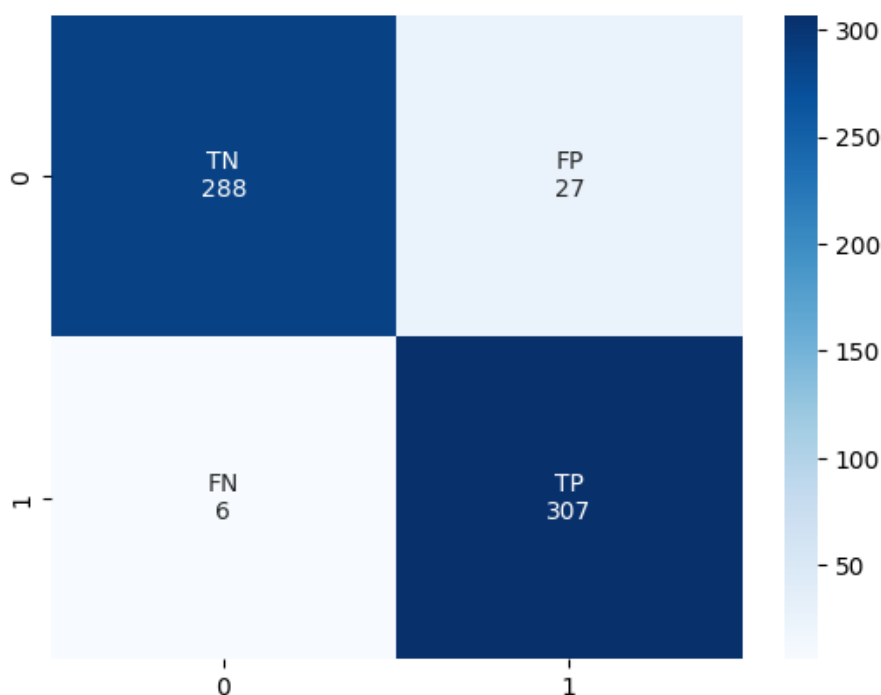
- Ma trận nhầm lẫn:

Ma trận nhầm lẫn là một bộ cục bảng cụ thể cho phép hình dung hiệu suất của một thuật toán và Là một trong những kỹ thuật đo lường hiệu suất phổ biến nhất và được sử dụng rộng rãi cho các mô hình phân loại.

4.3.2. Thử nghiệm

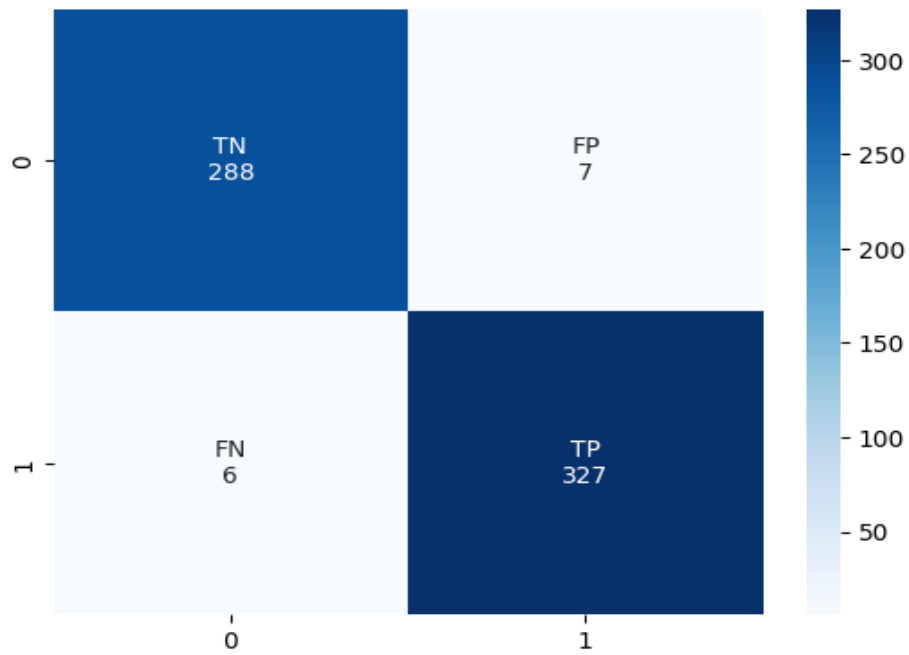
Chương trình được thử nghiệm bằng cách dự đoán 628 mẫu và thống kê bằng ma trận nhầm lẫn. Kết quả tương ứng với từng thuật toán sử dụng. Được mô tả bằng hình bên dưới

- Thuật toán KNN:



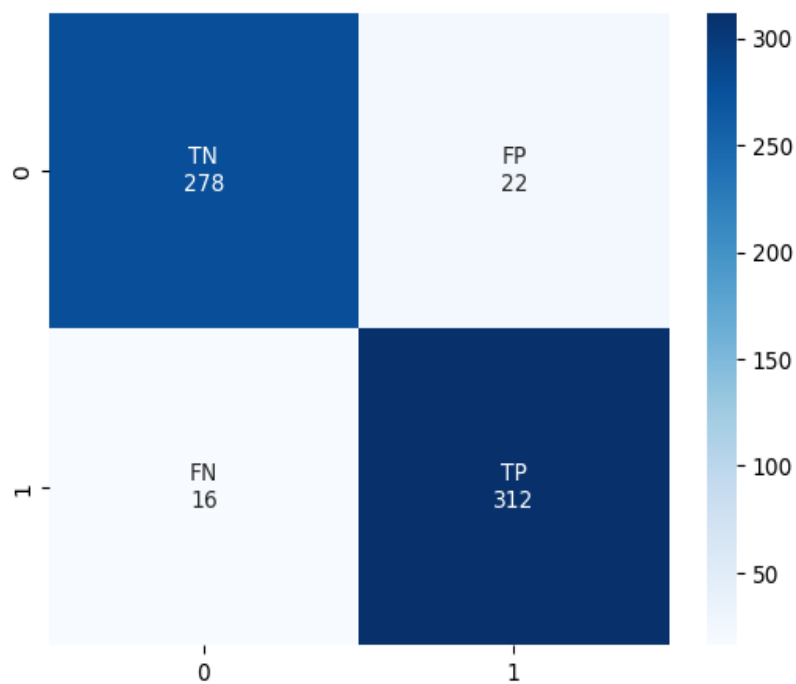
Hình 4.15: Ma trận nhầm lẫn khi dự đoán bằng thuật toán KNN

- Thuật toán SVM:



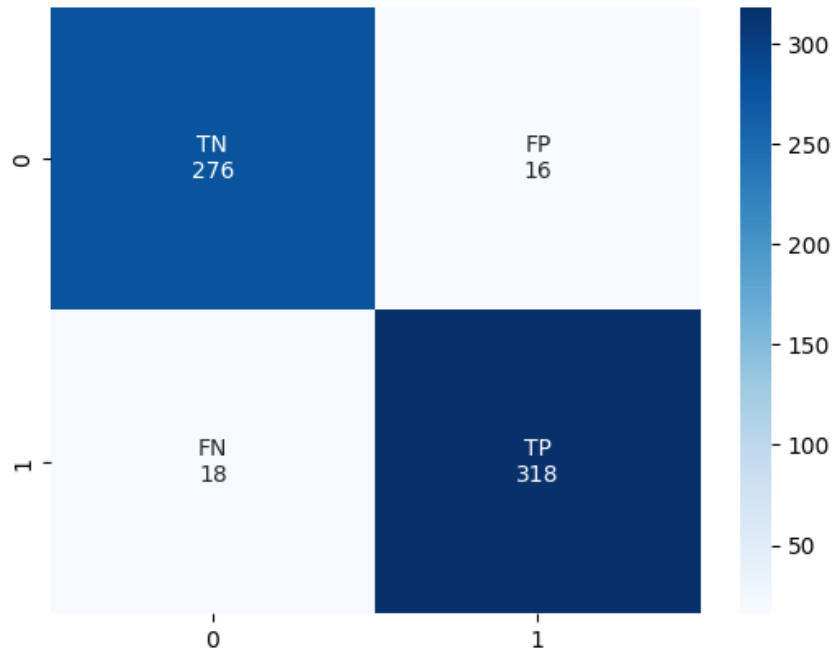
Hình 4.15: Ma trận nhầm lẫn khi dự đoán bằng thuật toán SVM

- Thuật toán NB:



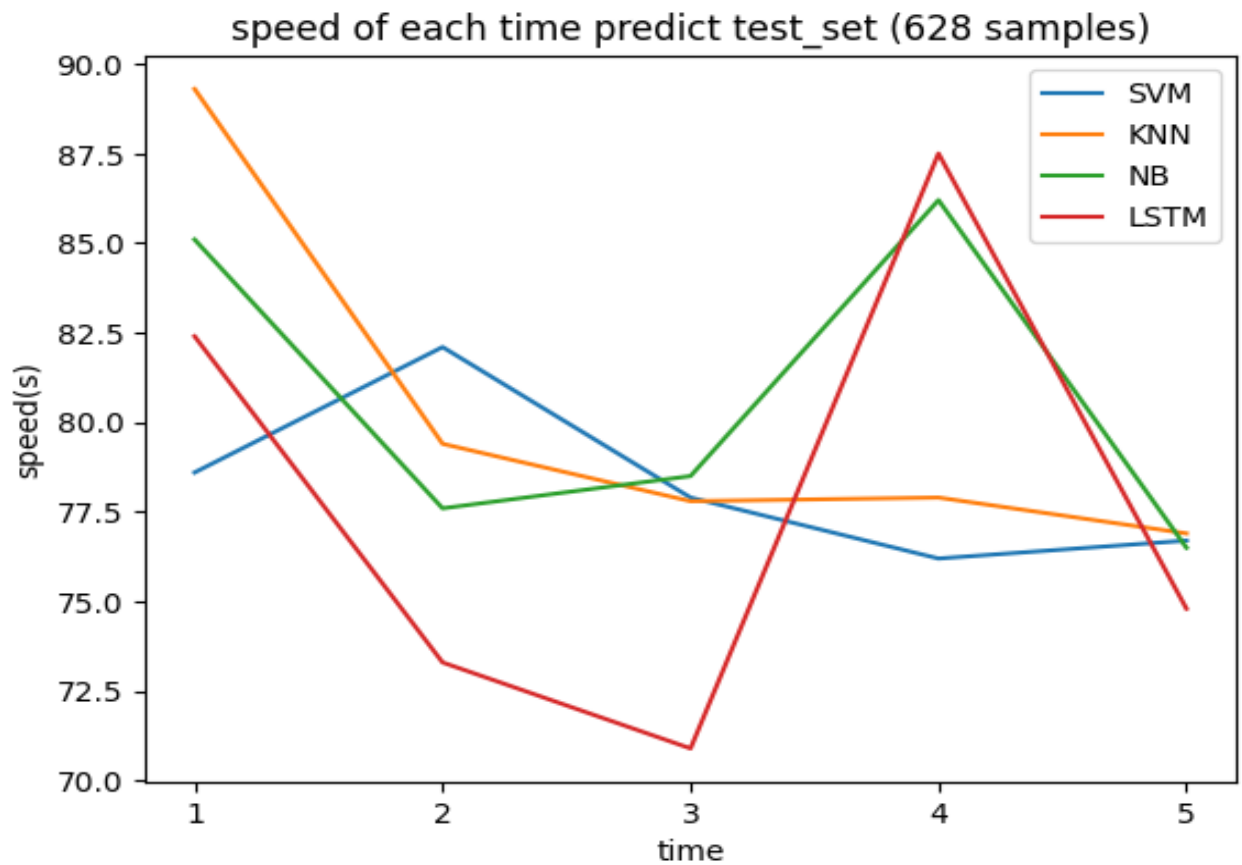
Hình 4.17: Ma trận nhầm lẫn khi dự đoán bằng thuật toán NB

- Thuật toán LSTM:



Hình 4.18: Ma trận nhầm lẫn khi dự đoán bằng thuật toán LSTM

Tốc độ của từng thuật toán cũng được đo lại như sau:



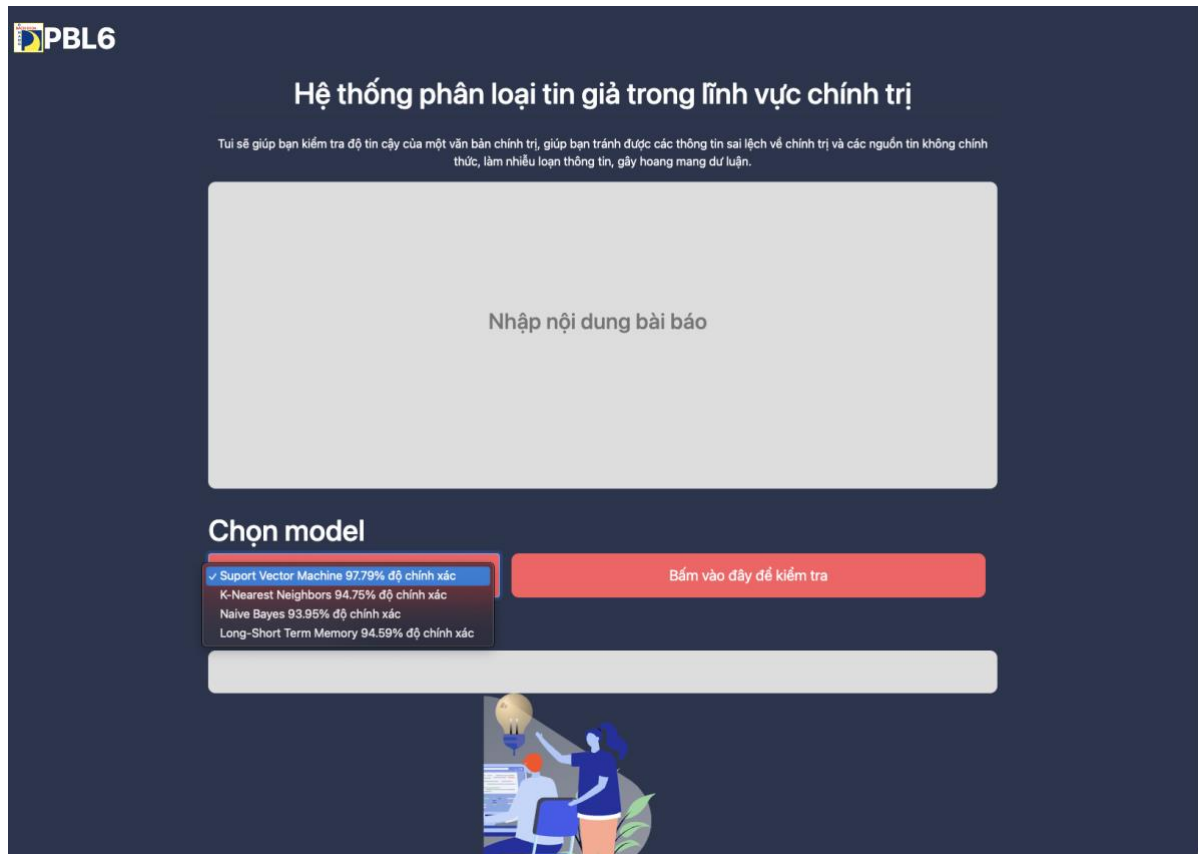
Hình 4.19: tốc độ của mỗi lần dự đoán của từng thuật toán

4.4. Ngôn ngữ cài đặt

Chương trình được cài đặt chủ yếu bằng Python để xây dựng mô hình và tạo API cho người dùng kết hợp với JavaScript và (HTML, CSS) để xây dựng giao diện website để mọi người có thể sử dụng.

4.5. Kết quả

4.5.1. Giao diện chính của chương trình



Hình 4.20: Giao diện của chương trình

4.5.2. Kết quả thực thi của chương trình

Mô tả:

Người dùng nhập văn bản cần nhận diện vào ô input sau đó chọn model (khuyến khích chọn cả 4 để có kết quả tốt) sau đó bấm nút bên cạnh để kiểm tra, hệ thống sẽ call API để mô hình tương ứng nhận dạng, kết quả sẽ được trả về ô kết quả ngay bên dưới nút bấm.

Hình 4.21: Kết quả khi thực thi chương trình

4.5.3. Nhận xét đánh giá

Bên cạnh những vấn đề đã đạt được, dự án còn một số vấn đề chưa giải quyết được, hoặc chưa tối ưu trong quá trình nghiên cứu. Cụ thể như sau:

- Chưa thể gán nhãn dữ liệu một cách nhanh, nhiều và hiệu quả.
- Lượng tin giả thu thập chủ yếu ở các trang chuyên tung tin giả nên phong cách viết bài thường giống nhau.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Kết quả đạt được

Đồ án này đã trình bày các kết quả nghiên cứu phân loại tin giả trong lĩnh vực chính trị dựa trên các mô hình học máy.

Kết quả đạt được là:

- Đã xây dựng thành công hệ thống phân loại tin giả trong lĩnh vực chính trị.
- Thử nghiệm các mô hình học máy để ứng dụng trong hệ thống phân loại tin giả trong lĩnh vực chính trị.

Dựa trên kết quả thử nghiệm đã so sánh các phương pháp mã hóa văn bản cũng như mô hình học máy trong việc phân loại tin giả trong lĩnh vực chính trị

5.1.2. Giới hạn của đồ án

Về cơ bản, hệ thống đã thực hiện hoàn thành các chức năng đặt ra là giúp người dùng có thể phân loại tin tin giả trong lĩnh vực chính trị. Tuy nhiên tập dữ liệu ban đầu chỉ ở mức thử nghiệm

Điểm hạn chế của đồ án là số chiều vectơ của văn bản chưa rút gọn một cách có hệ thống. Lượng dữ liệu thu thập vẫn chưa đủ phong phú.

5.2. Hướng phát triển

Phân loại tin giả là một hướng nghiên cứu đang được quan tâm đặc biệt là trong ngôn ngữ tiếng Việt còn nhiều vấn đề cần được quan tâm. Trong giới hạn nghiên cứu của đồ án nhóm xin đề xuất hướng nghiên cứu trong tương lai của đề tài này là.

- Mở rộng các lĩnh vực để có thể đa dạng trong việc nhận dạng tin giả
- Cải thiện độ chính xác để tăng độ tin cậy của dự án
- Thử nghiệm với các phương pháp học máy khác

TÀI LIỆU THAM KHẢO

1. Understanding LSTM Networks -- colah's blog
2. DeepLearning for Sentiment Analysis at Stanford
3. CS231n: Convolutional Neural Networks for Visual Recognition
4. Natural Language Processing Recipes - Akshay Kulkarni, Adarsha Shivanand Shivanand
5. Natural Language Processing with Python
6. Feature Extraction and Embeddings in NLP: A Beginners guide to understand Natural Language Processing
7. Text-based emotion detection: Advances, challenges, and opportunities
8. https://phamdinhhkhanh.github.io/deepai-book/ch_ml/SVM.html

PHỤ LỤC

Tất cả mã nguồn và cài đặt được lưu trữ tại.

<https://drive.google.com/drive/folders/1VipjSzVe4f7XTds34JkFf1sY2czGG>

NK