**Mẫu 1: TRANG BÌA**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN TUẤN HUY - 51703103**

**NGUYỄN ANH CHÂU - 51703051**

**Phân loại hình thái của các bình luận trên miền ngôn ngữ tiếng Việt**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**HỆ THỐNG THÔNG TIN**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN TUẤN HUY - 51703103**

**NGUYỄN ANH CHÂU - 51703051**

**Phân loại hình thái của các bình luận trên miền ngôn ngữ tiếng Việt**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**HỆ THỐNG THÔNG TIN**

Người hướng dẫn

**PGS-TS. Lê Anh Cường**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn PGS-TS Lê Anh Cường đã hướng dẫn cho chúng em các gợi ý cũng như là thầy luôn hỏi thăm và giúp đỡ chúng em rất nhiều. Việc báo cáo theo đúng tiến độ cũng góp phần hối thúc chúng em hoạt động và không lãng phí thời gian. Tại thời gian em viết báo cáo này thì dự án của em đã được hoàn thành. Một lần nữa trân trọng cảm ơn thầy.

*TP. Hồ Chí Minh, ngày 5 tháng 7 năm 2022*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Trần Tuấn Huy*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nguyễn Anh Châu*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Nhóm chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em và được sự hướng dẫn khoa học của PGS-TS Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào nhóm chúng em xin hoàn toàn chịu trách nhiệm về nội dung Báo cáo Dự án CNTT 2 của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 5 tháng 7 năm 2022*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Trần Tuấn Huy*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nguyễn Anh Châu*

**PHÂN LOẠI HÌNH THÁI CÁC BÌNH LUẬN**

**TRÊN MIỀN NGÔN NGỮ TIẾNG VIỆT**

**TÓM TẮT**

Trong dự án lần này, chúng em sẽ phận loại hình thái của các bình luận trên miền ngôn ngữ tiếng Việt. Các bình luận này được các người dùng trên mạng xã hội bình luận trong một bài viết nhằm nêu quan điểm, ý kiến cá nhân về bài viết đó. Sau khi thu thập bình luận, các bình luận này sẽ được đưa vào xử lý để chuẩn hóa và biến đổi để đưa vào mô hình. Sau đó dữ liệu sẽ được huấn liệu và đưa ra dự đoán cho dữ liệu khác. Cuối cùng là ta sẽ so sánh các mô hình học máy với nhau để có thể đưa ra nhận xét.

MỤC LỤC

**DANH MỤC HÌNH VẼ x**

**DANH MỤC BẢNG BIỂU xi**

**DANH MỤC CÁC CHỮ VIẾT TẮT xii**

**CHƯƠNG 1. MỞ ĐẦU 12**

1.1 Lý do chọn đề tài 12

1.2 Mục tiêu thực hiện đề tài 12

1.3 Đối tượng và phạm vi nghiên cứu 12

1.4 Phương pháp nghiên cứu 12

1.5 Ý nghĩa thực tiễn của đề tài 13

**CHƯƠNG 2. TỔNG QUAN 14**

2.1 Giới thiệu 14

*2.1.1 Python và Framework Flask 14*

*2.1.2 Angular framework 14*

2.2 Sơ lược về dự án 15

2.3 Các bài viết về phân tích cảm xúc miền ngôn ngữ tiếng Việt 15

*2.3.1 Phân loại tài liệu tiếng Việt - Trần Thị Thu Thảo và Vũ Thị Chinh 15*

*2.3.2 Phản hồi khách hàng – Phạm Hữu Quang 16*

*2.3.3 Dự đoán chủ đề của một bài báo – Nguyễn Văn Hiếu 17*

*2.3.4 So sánh với bản thân 18*

**CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 19**

3.1 Naive Bayes 19

3.2 Logistic Regression 20

3.3 Support Vector Machine 20

3.4 Long Short-Term Memory 21

**CHƯƠNG 4. PHƯƠNG PHÁP NGHIÊN CỨU 27**

4.1 Dataset 27

4.2 Kết quả 27

**CHƯƠNG 5. KẾT LUẬN 30**

**CHƯƠNG 6. CẢI TIẾN TRONG TƯƠNG LAI 32**

**TÀI LIỆU THAM KHẢO 33**

**PHỤ LỤC**

# DANH MỤC HÌNH VẼ

Hình 1.1: Sơ đồ luồng chạy của hệ thống 13

Hình 1.2: Từ sau khi được vector hóa 13

Hình 2.1: Mô hình N-gram mà tác giả Thảo và Chinh sử dụng 15

Hình 2.2: Công thức tính trọng số mà tác giả đã sử dụng 16

Hình 2.3: Sơ đồ hệ thống của tác giả 16

Hình 2.4: Phân bố cấu trúc data của tác giả Quang 17

Hình 2.5: Kết quả thu được từ các epoch đầu 17

Hình 2.6: Cách tác giả Hiếu tiền xử lý câu 17

Hình 3.1: Công thức của mô hình phân lớp NB 19

Hình 3.2: SVM tạo ra hyperplane 21

Hình 3.3: Module lặp trong RNN chuẩn chứa một tầng đơn 22

Hình 3.4: Module lặp trong LSTM chứa 4 tầng tương tác với nhau 22

Hình 3.5: Trạng thái tế bào giúp thông tin được giữ nguyên 23

Hình 3.6: Tầng mạng sigmoid và một phép nhân 23

Hình 3.7: Tham số đầu vào của tầng sigmoid cho ra [0, 1] [0,1] 24

Hình 3.8: Tầng sigmoid và tầng tanh để cập nhật trạng thái tế bào 24

Hình 3.9: Cập nhập trạng thái tế bào cũ 25

Hình 3.10: Quyết định đầu ra 25

Hình 3.11: LTSM của Gers & Schmidhuber (2000) 26

Hình 3.12: LSTM nối 2 cổng loại trừ và đầu vào 26

Hình 3.13: LSTM Gated Recurrent Unit (GRU) của Cho, et al. (2014) 26

Hình 4.1: Các trang trên facebook sẽ truy cập 27

Hình 4.2: Kết quả khi chạy mô hình 28

Hình 4.3: Kết quả thực tế 28

Hình 5.1: Độ quan trọng của việc tiền xử lý câu 30

# DANH MỤC BẢNG BIỂU

Bảng 1.1: Biến đo lường 5

Bảng 1.2: Ví dụ 5

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| NLP  LR | Natural Language Processing  Logistic Regression |
| SVM | Support Vector Machine |
| NB | Naive Bayes |
| LSTM | Long Short-term memory |
| TF-IDF  DOM  CNTT | Term frequency-inverse document frequency  Document Object Model  Công Nghê Thông Tin |

CHƯƠNG 1 – MỞ ĐẦU

* 1. Lý do chọn đề tài

Tiếng Việt từ lâu đã được xem là rất phức tạp không chỉ về mặt ngữ pháp, ý nghĩa mà còn về cả mặt hình thức. Chính vì thế mà việc xác định cảm xúc của một người trong câu nói cũng khá là khó khăn. Với mong ước có thể sử dụng học máy vào tiếng Việt để có thể dự đoán cảm xúc của người dùng một cách tự động thì chúng em thực hiện đề tài này. Một phần có thể dự đoán cảm xúc, một phần có thể áp dụng được mô hình học máy vào tiếng Việt vì đây là một chủ đề khá hay do sự đa dạng của tiếng Việt nhưng lại có rất ít bài viết được tìm thấy.

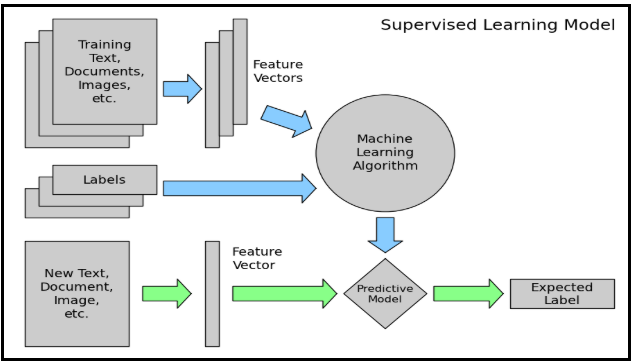
* 1. Mục tiêu thực hiện đề tài

Trong dự án lần này, chúng em sẽ hướng đến việc xử lý các câu đầu vào sao cho phù hợp để đưa vào mô hình. Kết quả cuối cùng thu được sẽ là nhãn (cảm xúc) dự đoán của câu được đưa vào. Sau đó là sự so sánh giữa các mô hình học máy với nhau, đánh giá mức độ chính xác cũng như đưa ra nhận xét cho các mô hình được sử dụng.

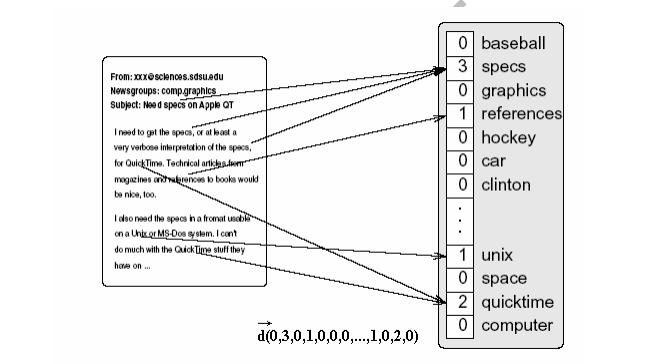
* 1. Đối tượng và phạm vi nghiên cứu

Đối tượng được nhắm đến trong dự án lần này là những bình luận của người dùng ở trên các bài viết của các trang tin facebook. Các trang tin này là các trang tin lớn có rất nhiều lượt bình luận cũng như là tương tác. Các bài viết sẽ viết về một chủ đề hay một điều gì đó cần bàn luận và phía dưới là ý kiến cũng như là cảm xúc của người dùng về thông tin mà bài viết muốn truyền tải.

* 1. Phương pháp nghiên cứu
* Sử dụng selenium để giả lập trình duyệt Chrome để thu thập các bình luận.
* Các bình luận sẽ được đánh nhãn tương ứng, sau được sử lý chuẩn hóa bằng regex, loại bỏ stopword và vector hóa bằng cách sử dụng TF-IDF
* Sau khi xử lý dataset thì tiến hành chia tập train và test với tỉ lệ 85:15.
* Các mô hình học máy bao gồm SVM, LR, NB và LSTM
* Sau khi train thì ta sẽ lưu mô hình ra bên ngoài, tiếp tục sử dụng tập test để kiểm tra độ chính xác.
* Các thông tin đánh giá bao gồm tỷ lệ chính xác và thời gian mô hình chạy.
* Cuối cùng là thử nghiệm bằng cách nhập một câu bất kỳ vào hệ thống xem mô hình trả ra nhãn như thế nào.



Hình 1.1: Sơ đồ luồng chạy của hệ thống



Hình 1.2: Từ sau khi được vector hóa

* 1. Ý nghĩa thực tiễn của đề tài

Dự án được thực hiện nhằm vào hai mục đích chính:

* Giúp dự đoán được cảm xúc trong một câu
* Giúp thu thập được bình luận người dùng, nhằm đánh giá xu hướng cảm xúc mọi người cũng như là điều chỉnh hành vi, bài viết của các trang mạng sao cho hợp lý, phù hợp với bối cảnh xã hội.

CHƯƠNG 2 – TỔNG QUAN

*2.1 Giới thiệu*

TypeScrpit và Python là 2 ngôn ngữ lập trình chính được sử dụng trong môn học dự án Công nghệ thông tin 2 này:

2.1.1 Python và Framework Flask:

Phân tích cảm xúc (Sentiment Analysis) là một ứng dụng trí tuệ nhân tạo, nó sử dụng các thuật toán phức tạp để xử lý ngôn ngữ tự nhiên của con người (NLP) và xác định các đặc điểm cảm xúc tiêu cực/tích cực tại một thời điểm thông qua văn bản hoặc lời nói. Các nguồn dữ liệu được phân tích phổ biến như Social media, Blog, Website đánh giá sản phẩm, tổng đài Contact center, …

Có 2 dạng đo lường cảm xúc:

– Thông qua văn bản: Xác định các từ, cụm từ có liên quan đến cảm xúc, hành vi.

– Thông qua lời thoại: Xác định các từ, cụm từ, đặc điểm giọng nói (cường độ, cao độ, nhịp điệu, hơi thở, …) có liên quan đến cảm xúc, hành vi.

Python là ngôn ngữ mãnh mẽ bao gồm nhiều thư viện (nổi bật NLTK) hỗ trợ cho việc học máy, phân tích cảm xúc nên được sử dụng để xử lý từ khâu crawl data, xử lý data, training, tính toán xác xuất cho ra kết quả cuối cùng.

2.1.2 Angular framework:

Angular là Framework được phát triển để xây dựng các Single Page Application (SPA) sử dụng HTML, TypeScript … Nó triển khai chức năng cốt lõi và các chức năng tùy chọn như một tập hợp các thư viện TypeScript mà chúng ta import vào các ứng dụng của mình.

Angular là một Framework dựa trên mô hình Model-View-Controller (MVC). Nó cung cấp hướng dẫn rõ ràng về cách ứng dụng nên được cấu trúc và cung cấp luồng dữ liệu hai chiều trong khi cung cấp DOM thực. Một ứng dụng Angular bao gồm các phần: modules, components, templates, metadata, services, dependency injection.

Với các đặc điểm trên, Angular framework được chúng em sử dụng để làm Front End cho Web app.

*2.2 Sơ lược về dự án*

Facebook ngày càng phổ biến trong đời sống giới trẻ, và việc xác định được cảm xúc của người dùng thông qua lời nói cực kỳ quan trọng, admin: data thu thập được trên facebook để biết được thông tin điều chỉnh hành vi cũng như xác định hướng tiếp cận với các người dung khác.

Sơ lược về dataset gồm hơn 6500 bình luận thu thập kèm nhãn được đánh, các nhãn gồm Giận dữ, :S, :))))), :(((((, Yêu, Ngạc nhiên, Rác,…

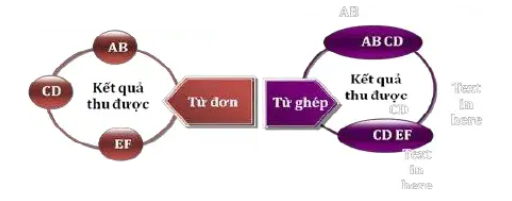
Vấn đề còn tồn tại là viết tắt, chửi bậy, sai chính tả, không viết dấu, nhiều từ mới chưa được update, tiếng lóng.

Vấn đề tập trung nghiên cứu: biểu cảm trong câu. Trong dự án lần này chúng em sẽ tập trung phân loại nhãn cho câu đã được xử lý cũng như là áp dụng việc vector hóa từ để nâng cao hiệu quả cho mô hình. Từ đó có thể dự đoán chính xác hơn nhãn cho một câu ngẫu nhiêu được đưa vào ứng dụng.

2.3 Các bài viết về phân tích cảm xúc miền ngôn ngữ tiếng Việt

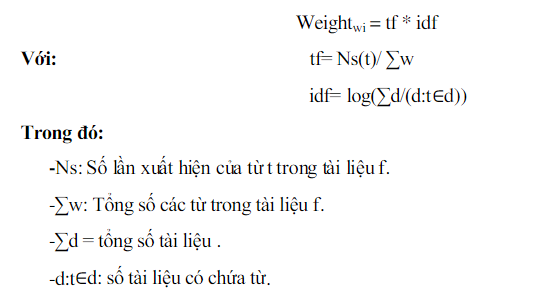
2.3.1 Phân loại tài liệu tiếng Việt - Trần Thị Thu Thảo và Vũ Thị Chinh

Dataset mà tác giả sử dụng là các bài báo khoa học điện tử thuộc lĩnh vực CNTT (thu thập từ nhiều nguồn khác nhau). Về việc tách từ thì tác giả sử dụng mô hình N-gram, với n thường ở ngưỡng 1 và 2.

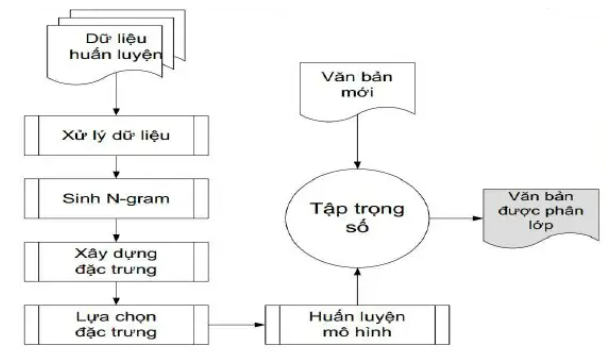


Hình 2.1: Mô hình N-gram mà tác giả Thảo và Chinh sử dụng

Về việc vector từ và tính trọng số, tác giả sử dụng TF-IDF. Các bước thực thiện gồm loại bỏ stopword, đếm tần suất xuất hiện và tính trọng số của từ. Thuật toán mà tác giả đã sử dụng là NB và SVM.



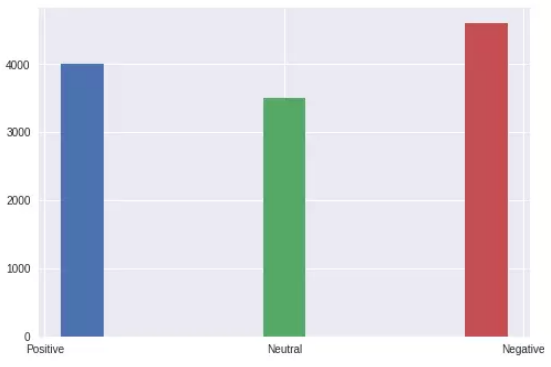
Hình 2.2: Công thức tính trọng số mà tác giả đã sử dụng



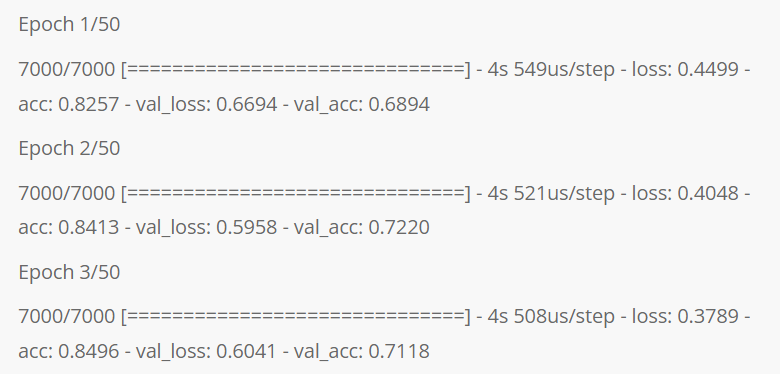
Hình 2.3: Sơ đồ hệ thống của tác giả

2.3.2 Phản hồi khách hàng – Phạm Hữu Quang

Dataset mà tác giả sử dụng là từ các trang web và fanpage của nhà hàng, gồm 3 loại nhãn là Negative, Positive và Neutral. Trong giai đoạn tiền xử lý thì tác giả sẽ tách từ bằng thư viện underthesea, loại bỏ các ký tự đặc biệt và chuẩn hóa các từ viết tắt hoặc đồng âm. Tác giả sử dụng Word2Vec của gensim để vector hóa dữ liệu thành 128 chiều và train mô hình bằng CNN với epoch là 50.



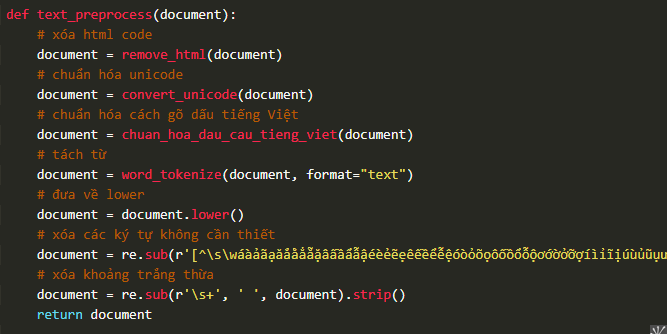
Hình 2.4: Phân bố cấu trúc data của tác giả Quang



Hình 2.5: Kết quả thu được từ các epoch đầu

2.3.3 Dự đoán chủ đề của một bài báo – Nguyễn Văn Hiếu

Với data thì tác giả thu thập ở binhvq/news-corpus. Về bước tiền xử lý data thì tác giả hiếu làm tương tự như tác giả Phạm Hữu Quang. Thuật toán tác giả hiếu đã sử dụng bao gồm NB, LR, SVM và Fasttext.



Hình 2.6: Cách tác giả hiếu tiền xử lý câu

2.3.4 So sánh với bản thân

Dataset của chúng em được thu thập ở các fanpage facebook nhờ vào thư viện selenium cũng như tham khảo cách sử dụng trên youtube. Cách tiền xử lý data đầu vào cũng tương tự như tác giả Nguyễn Văn Hiếu và Phạm Hữu Quang nhưng không có phần loại bỏ text HTML hoặc link vì các bình luận này toàn là quảng cáo nên sẽ bị loại bỏ.

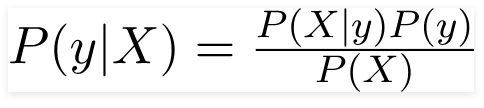
Các thuật toán em đã áp dụng vào đề tài bao gồm LR, SVM, NB và LSTM, sử dụng N-gram với n ở ngưỡng 1 và 2. Để vector hóa câu em dùng TF-IDF và đưa vào mô hình.

CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT

Học máy cho NLP yêu cầu tiền xử lý để trích xuất các tính năng từ (các) văn bản gốc. Cụ thể, nó yêu cầu chuyển đổi dữ liệu văn bản sang biểu diễn số trước khi các thuật toán có thể xử lý chúng, tức là vectơ hóa. Trong đề tài này chúng em sử dụng 4 thuật toán để so sánh độ chính xác, sự hiệu quả giữa 4 thuật toán này.

*3.1 Naive bayes*

Bộ phân loại NB sử dụng xác suất thống kê để đưa ra dự đoán dựa trên kiến thức trước đó về các điều kiện có thể liên quan. Nói cách khác, nó sử dụng các xác suất có điều kiện của mỗi đặc điểm từ vựng xảy ra trong văn bản khẳng định hoặc phủ định trong dữ liệu đào tạo để đi đến kết quả.



Hình 3.1: Công thức của mô hình phân lớp NB

* P(y|X) posterior probability: xác suất của y với điều kiện có X
* P(X|y) likelihood: xác suất của X khi đã biết y
* P(y) prior probability của y
* P(X) prior probability của X

Đối với NB, ta tạo một DTM đơn giản (ma trận thuật ngữ tài liệu) để dùng vào mô hình. Đầu tiên là làm sạch dữ liệu để giúp giảm kích thước và mô hình (bộ phân loại) hiệu suất. Sau đó biểu diễn vectơ được xây dựng bằng cách đếm TF (Tần suất kỳ hạn) và tính trọng số của chúng với IDF (Tần suất tài liệu nghịch đảo). Chúng ta có thể sử dụng N-gram (các thuật ngữ liên tiếp) để nắm bắt một số ngữ cảnh trong văn bản chẳng hạn như phủ định. N-gram là một công không phải lúc nào cũng chính xác, điều này đôi lúc lại tác động tiêu cực lên mô hình của chúng ta.

Mô hình NB sử dụng dễ dàng và có tốc độ xử lý khá nhanh, hỗ trợ **các** nỗ lực Phân tích Cảm xúc **quy mô lớn** vì quá trình tính toán cần nhanh chóng. NB có một số nhược điểm là rất phụ thuộc vào dữ liệu đầu vào (phải đầy đủ và mang tính đại diện). Việc thiếu dữ liệu đào tạo tốt dẫn đến suy luận kém. Ngoài ra nó cũng có thể bị ảnh hưởng nếu có sự sai lệch hoặc mất cân bằng trong dữ liệu. Thuật toán này thường được dùng trong phân tích sắc thái, lọc thư rác, gợi ý, …

Một số kiểu mô hình NB:

* Multinomial NB: Thường được dùng trong phân loại văn bản
* Bernoulli NB: Thường được sử dụng khi đầu vào là giá trị nhị phân 0 hoặc 1
* Gausian NB: Thường được dùng khi X nhận giá trị liên tục

*3.2 Logistic Regression*

LR là một phương pháp phân tích thống kê được dùng để dự đoán giá trị dữ liệu dựa trên các quan sát trước đó của tập dữ liệu. Mục đích của LR là ước tính xác suất của các sự kiện, bao gồm xác định mối quan hệ giữa các tính năng từ đó đự đoán xác suất của các kết quả, nên đối với LR ta sẽ có:

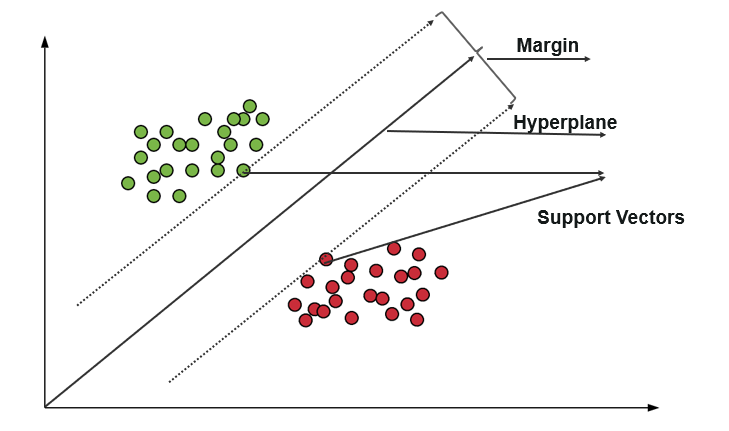
* **Input**: dữ liệu input (ta sẽ coi có hai nhãn là 0 và 1).
* **Output**: Xác suất dữ liệu input rơi vào nhãn 0 hoặc nhãn 1.

Về mặt toán học thì LR được xem là một hàm nhiều Linear Regression. Khi chọn mô hình cho phân tích hồi quy logistic, một điều quan trọng khác cần xem xét là sự phù hợp của mô hình. Việc thêm các biến độc lập vào mô hình hồi quy logistic sẽ luôn làm tăng số lượng phương sai được giải thích trong tỷ lệ cược log (thường được biểu thị bằng R²). Tuy nhiên, việc thêm ngày càng nhiều biến vào mô hình có thể dẫn đến trang bị quá mức, điều này làm giảm khả năng tổng quát hóa của mô hình ngoài dữ liệu mà mô hình phù hợp. Biến phụ thuộc X phải có tính chất lưỡng phân (ví dụ: có hoặc không).

Các bài toán thường ứng dụng LR: xác định nguy cơ gây bệnh, phân loại từ, dự báo thời tiết, bình chọn, …

*3.3 Support Vector Machine*

SVM là một mô hình phân loại đào tạo dữ liệu dưới dạng các điểm, xây dựng một siêu phẳng (hyperplane) có (n - 1) chiều trong không gian n chiều của dữ liệu sao cho phân loại các lớp tối ưu nhất. Ở không gian 2 chiều thì siêu phẳng này là 1 đường thẳng phân cách chia mặt phẳng không gian thành 2 phần tương ứng 2 lớp với mỗi lớp nằm ở 1 phía của đường thẳng.



Hình 3.2: SVM tạo ra **hyperplane**

SVM dùng một tập hợp con các điểm huấn luyện trong hàm quyết định, có hiệu quả về bộ nhớ và trong các không gian có chiều cao. Nhược điểm của SVM là thuật toán không cung cấp trực tiếp các ước lượng xác suất.

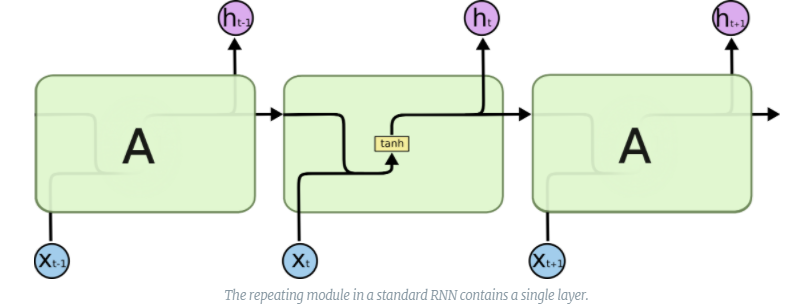
SVM thường được dùng trong kinh doanh để so sánh hiệu suất của cổ phiếu trong một khoảng thời gian.

*3.4 Long Short-Term Memory*

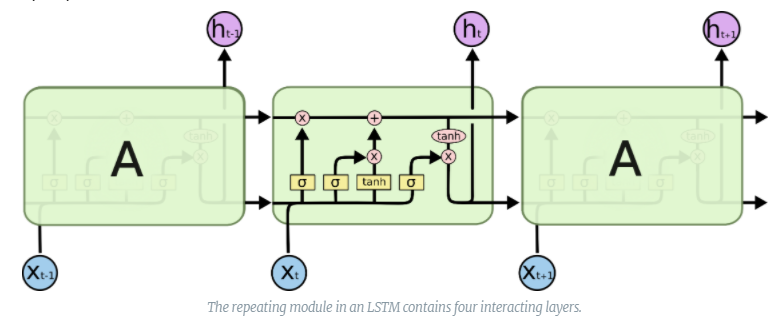
LSTM là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa, được [Hochreiter & Schmidhuber giới thiệu (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf). LSTM cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được.

LSTM cũng có kiến trúc dạng chuỗi các module lặp đi lặp lại như RNN chuẩn (các module này có cấu trúc rất đơn giản), nhưng các module trong nó có cấu trúc khác. Thay vì chỉ có một tầng mạng neuron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.

****

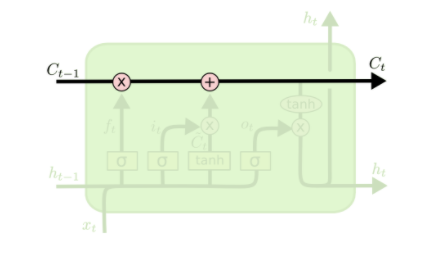
Hình 3.3: Module lặp trong RNN chuẩn chứa một tầng đơn.



Hình 3.4: Module lặp trong LSTM chứa 4 tầng tương tác với nhau.

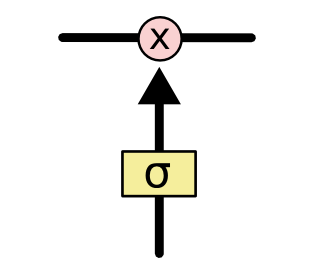
Ở hình trên, mỗi một đường là một vector giữa hai nút. Các nút màu hồng đại diện cho các phép toán vector, các ô màu vàng dùng để học trong từng mạng neuron. Các đường hợp nhau kí hiệu việc kết hợp, các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới nơi khác.

Chìa khóa của LSTM là trạng thái tế bào (cell state) - đường chạy thông ngang, là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



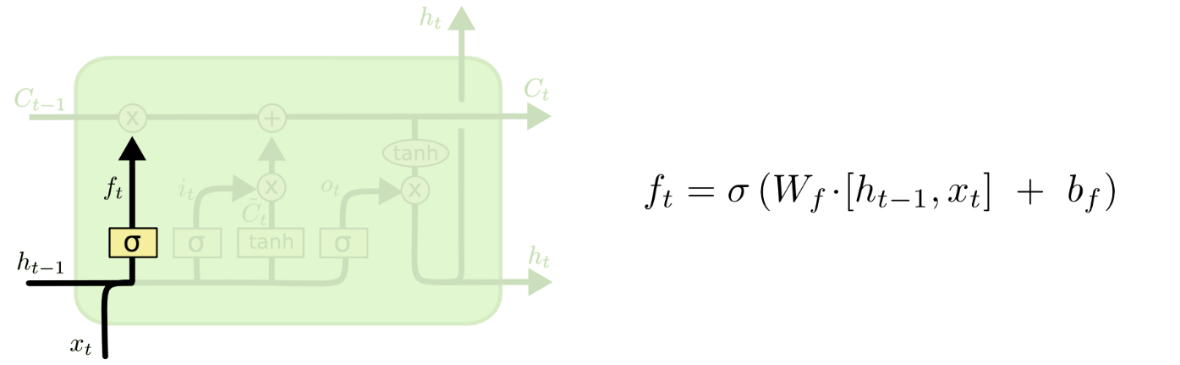
Hình 3.5: Trạng thái tế bào giúp thông tin được giữ nguyên

Các thông tin cần thiết cho trạng thái tế bào có thể điều chỉnh được bởi các cổng (gate). Đây là nơi sàng lọc thông tin, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân. Tầng sigmoid sẽ cho đầu ra là một số trong khoảng [0, 1] [0,1], với 00 là không có thông tin đi qua còn 11 là cho tất cả đi qua. Một LSTM gồm có 3 cổng như vậy.



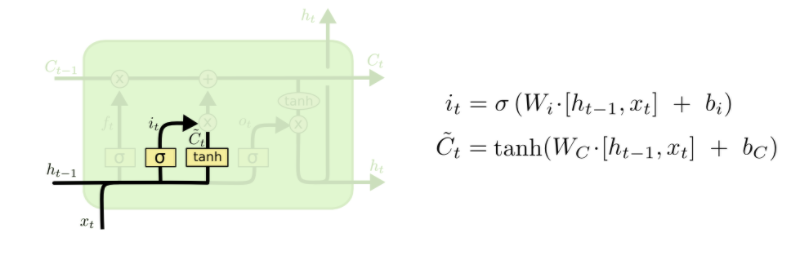
Hình 3.6: Tầng mạng sigmoid và một phép nhân

Đầu tiên LSTM quyết định thông tin cần bỏ đi từ trạng thái tế bào, do tầng sigmoid đảm nhiệm (forget gate layer). Đầu vào sẽ là h\_{t-1}*ht*−1​ và x\_t*xt*​ rồi đưa ra kết quả trong khoảng [0, 1] [0,1] cho mỗi số trong trạng thái tế bào C\_{t-1} *Ct*−1​.



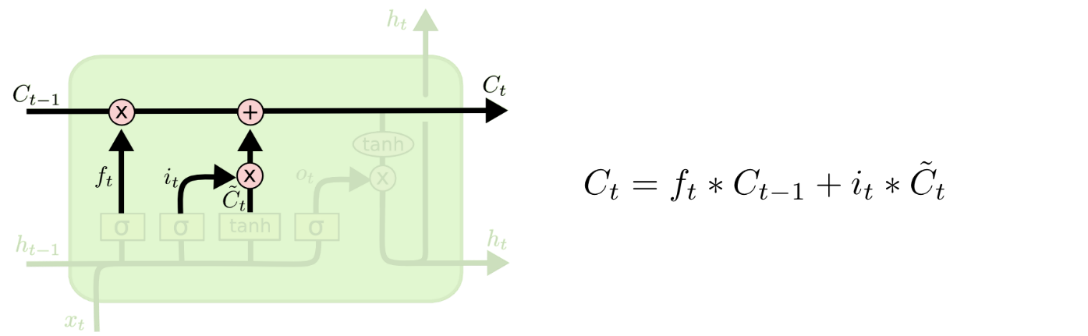
Hình 3.7: Tham số đầu vào của tầng sigmoid cho ra [0, 1] [0,1]

Tiếp theo LSTM quyết định thông tin nào lưu vào trạng thái tế bào. Đầu tiên là sử dụng một tầng sigmoid (input gate layer) để quyết định giá trị sẽ cập nhập. Tiếp theo là một tầng *tanh* tạo ra một vector cho giá trị mới \tilde{C\_t} *Ct*​~​ nhằm thêm vào cho trạng thái rồi kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.



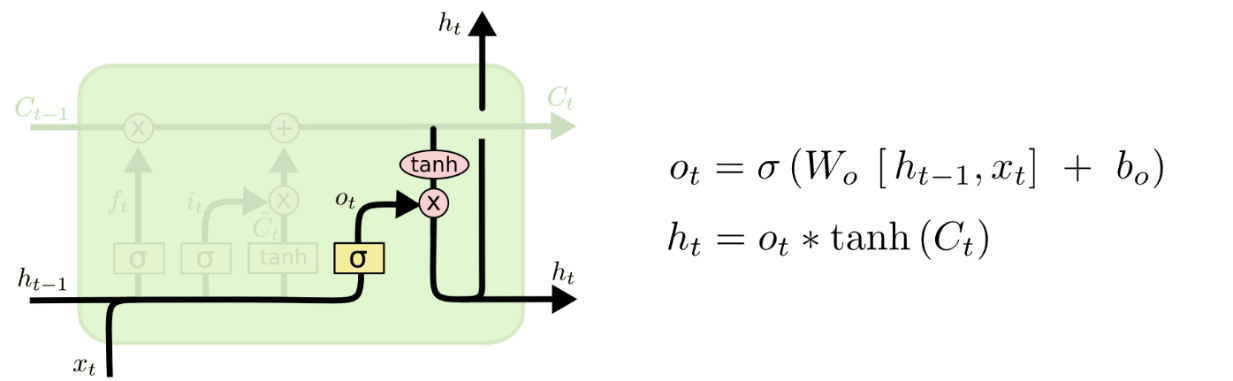
Hình 3.8: Tầng sigmoid và tầng *tanh* để cập nhật trạng thái tế bào

Để cập nhập trạng thái tế bào cũ C\_{t-1} *Ct*−1​ thành trạng thái mới C\_t *Ct*, ta sẽ nhân trạng thái cũ với f\_t *ft*​ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm i\_t \* \tilde{C\_t} *it*​∗*Ct*​~​. Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.



Hình 3.9: Cập nhập trạng thái tế bào cũ

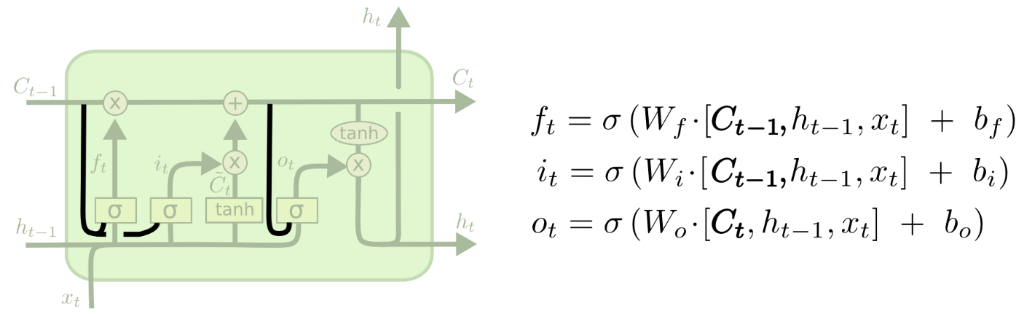
Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm *tanh* để có giá trị nó về khoảng [-1, 1] [−1,1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.



Hình 3.10: Quyết định đầu ra

Đây là một LSTM chuẩn thông thường. Nhưng trên thực tế, các bài báo về LTSM đều sử dụng một phiên bản hơi khác, giúp giải quyết phần nào đó trong cấu trúc của LTSM.

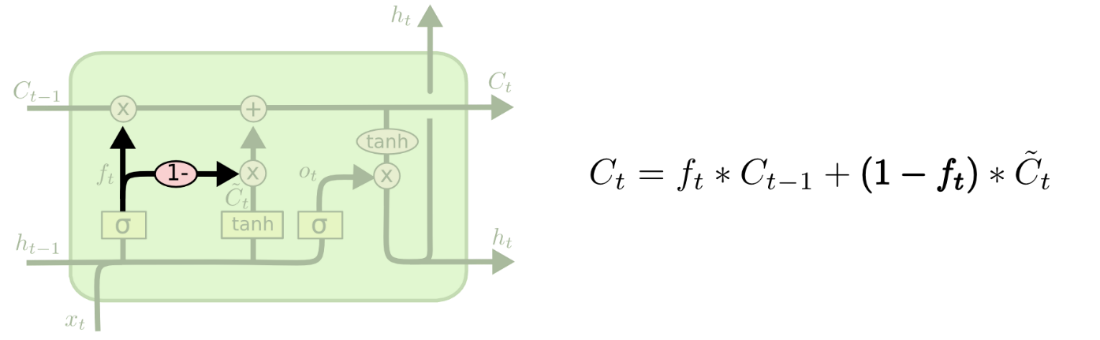
Một dạng LTSM phổ biến được giới thiệu bởi [Gers & Schmidhuber (2000)](ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf) được thêm các đường kết nối “peephole connections”, làm cho các tầng cổng nhận được giá trị đầu vào là trạng thái tế bào.



Hình 3.11: LTSM của [Gers & Schmidhuber (2000)](ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf)

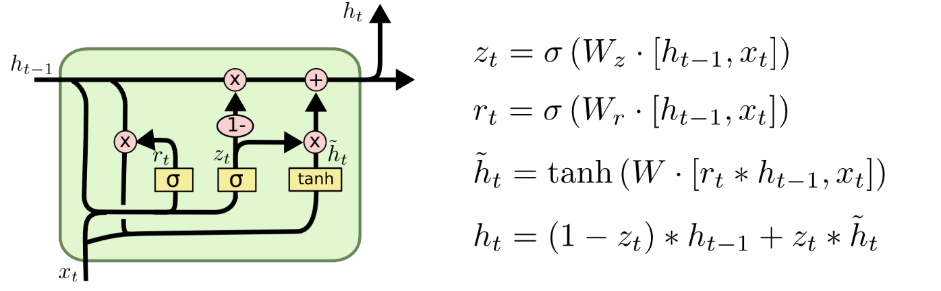
Hình trên mô tả các đường được thêm vào mọi cổng, nhưng cũng có những bài báo chỉ thêm cho một vài cổng mà thôi.

Một biến thể khác là nối 2 cổng loại trừ và đầu vào với nhau. Thay vì phân tách các quyết định thông tin loại trừ và thông tin mới thêm vào, ta sẽ quyết định chúng cùng với nhau luôn. Ta chỉ bỏ đi thông tin khi mà ta thay thế nó bằng thông tin mới đưa vào. Ta chỉ đưa thông tin mới vào khi ta bỏ thông tin cũ nào đó đi.



Hình 3.12: LSTM nối 2 cổng loại trừ và đầu vào

Một biến thể khá thú vị khác của LSTM là Gated Recurrent Unit, hay GRU được giới thiệu bởi [Cho, et al. (2014)](http://arxiv.org/pdf/1406.1078v3.pdf). Nó kết hợp các cổng loại trừ và đầu vào thành một cổng “cổng cập nhập” (update gate). Nó cũng hợp trạng thái tế bào và trạng thái ẩn với nhau tạo ra một thay đổi khác. Kết quả là mô hình của ta sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.



Hình 3.13: LSTM Gated Recurrent Unit (GRU) của [Cho, et al. (2014)](http://arxiv.org/pdf/1406.1078v3.pdf)

CHƯƠNG 4 – PHƯƠNG PHÁP NGHIÊN CỨU

*4.1 Dataset*

Crawl data từ các bài viết của các trang trên facebook và lưu các bình luận về một file txt. Có tổng cộng hơn 6500 bình luận. Sau khi xử lý loại bỏ đi các hàng trống cũng như là bình luận chứa quảng cáo (link mua hàng, cá độ) thì ta có 4497 bình luận được chia vào 7 nhãn gồm Giận dữ, :S, :))))), :(((((, Yêu, Ngạc nhiên, Rác. Trong số này có những bình luận không mang ý nghĩa về mặt cảm xúc cũng như những bình luận quá dài và chứa link quảng cáo thì sẽ bị loại bỏ.

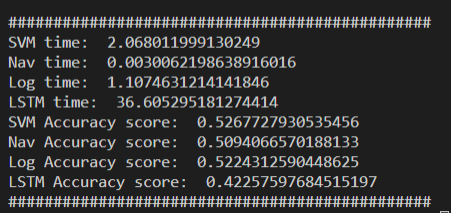
Dataset sẽ được tiền xử lý để có thể chuẩn hóa tiếng Việt cũng như loại bỏ stopword. Sau đó dataset sẽ được vector hóa trước khi đưa vào mô hình train. Dataset được chia theo tỉ lệ 85% cho tập train và 15% cho tập test. Sau khi train mô hình xong thì ta sẽ dùng tập test để kiểm tra và đánh giá sơ bộ về các thuật toán và các mô hình được train.



Hình 4.1: Các trang trên facebook sẽ truy cập

*4.2 Kết quả*

Kết quả được trả ra bao gồm thời gian để huấn luyện mô hình cho dataset đã qua xử lý và độ chính xác. Độ chính xác được tính bằng cách sử dụng mô hình để dự đoán nhãn của tập test. Sau đó so sánh nhãn ban đầu của tập test với lại nhãn đã được dự đoán. Thời gian được tính theo đơn vị s (giây).



Hình 4.2: Kết quả khi chạy mô hình



Hình 4.3: Kết quả thực tế

Dựa trên kết quả thực tế chúng em đã kiểm tra trên mô hình nhiều lần, mô hình đem đến kết quả chính xác nhất là LSTM. Nhìn sơ qua về bảng thống kê ta thấy NB cho kết quả cao nhất nhưng trên thực tế lại dự đoán sai nhiều nhất. SVM tuy có cao hơn LR nhưng trên thực tế thì 2 mô hình này đều giống như nhau và đều cho kết quả kém hơn LSTM. Các mô hình này thường dự đoán chính xác nhất ở các câu ngắn và dựa trên tần suất của từ trong câu đầu vào. Sở dĩ tỉ lệ của LSTM thấp hơn các mô hình khác là vì dataset ít và rất hỗn loạn, một phần khác vì cách học của LSTM nó phức tạp hơn các thuật toán còn lại.

Nếu đem các mô hình này áp dụng với các dataset khác có số lượng nhãn ít hơn (2 nhãn thay vì 7 nhãn như dataset hiện tại), số lượng đầu vào nhiều hơn (48000 so với 4497) có cấu trúc câu ổn định hơn thì tỷ lệ có thể lên đến hơn 80% và dự đoán khá chính xác.

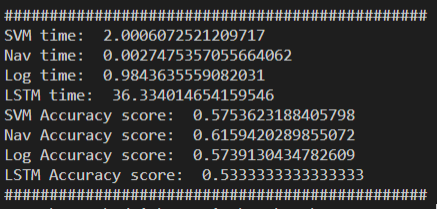
Một lý do khác khiến cho các mô hình ngay lúc này đây có tỉ lệ khá thấp có thể là việc xử lý dữ liệu đầu vào kém. Việc tinh chỉnh cũng như chuẩn hóa tiếng Việt một cách tối sẽ giúp cho tỉ lệ tăng lên rất nhiều. Việc xử lý dữ liệu kém dẫn đến hậu quả là dư thừa quá nhiều từ nhiễu, kèm với lượng dữ liệu ít góp phần làm cho việc đo tần suất từ bị hạn hẹp dẫn đến dự đoán nhãn sai.

CHƯƠNG 5 – KẾT LUẬN

Trong bài toán lần này, chúng em nghĩ rằng 2 giai đoạn quan trọng nhất của bài toán là tiền xử lý dữ liệu và chọn dataset. Việc lựa chọn data rất quan trọng đến sự thành bại của bài toán. Dataset cần có độ hỗn loạn thấp, giúp dễ dàng cho việc tiền xử lý câu cũng như huấn luyện mô hình. Nếu như đã chọn dataset không quá đẹp thì việc tiền xử lý data lại càng phải được đặt lên hàng đầu.



Chỉ với một dòng nhỏ như vậy cũng có thể làm tăng tỉ lệ chính xác của mô hình một cách đáng kinh ngạc.



Hình 5.1: Độ quan trọng của việc tiền xử lý câu

Tiếp theo là độ chính xác đến khinh ngạc của LSTM. Đây là một mô hình khá mạnh và cũng rất phổ biến hiện nay. Về độ hiệu quả LSTM không phải tuyệt đối nhưng tốt hơn nhiều so với các mô hình học máy thông thường.

Cuối cùng là không thể nhìn bề ngoài mà đánh giá được một mô hình là tốt hay tệ. Đối với hình 25, ta có thể thấy được sau khi thêm một bước nhỏ ở khâu tiền xử lý thì tỉ lệ đã tăng rất nhiều. Nhưng trên thực tế thì bước xử lý kia chưa hoàn toàn tối ưu, dẫn đến việc dữ liệu đầu vào đã sai hơn rất nhiều so với trước. Khi nhập một câu vào ứng dụng thì kết quả cho ra ở model hình 23 chính xác hơn rất nhiều. Có thể tỉ lệ ở hình 25 cao là do sự chênh lệch giữa các nhãn quá lớn (nhãn :))))) chiếm hơn 40%). Nếu nó dự đoán tất cả là :))))) thì nó đã cầm chắc trong tay hơn 40%, và trong hơn 50% còn lại nó chỉ dự đoán đúng hơn 10%. Điều này đúng là rất tệ.

CHƯƠNG 6 – CẢI TIẾN TRONG TƯƠNG LAI

Trong tương lai, việc cải thiện mô hình là rất cần thiết, thể hiện sự cầu tiến cũng như phát triển mô hình thực tế. Dựa vào các thiếu sót trong hiện tại, chúng em cần cải tiến như sau:

* Cần thu thập thêm để dataset đủ lớn, phục vụ cho mục đích huấn luyện.
* Do là data thu thập trên facebook nên không thể tránh được những bình luận vô nghĩa do đứt quãng (người dùng bình luận bằng hình ảnh nhưng con bot chỉ thu thập chữ). Cần cải tiến thêm để khắc phục điều này.
* Liên tục xem xét về độ cân bằng giữa các nhãn trong dataset, luôn đảm bảo rằng mô hình có thể học đủ để dự đoán chính xác.
* Tiếp tục cải tiến phần tiền xử lý dữ liệu, các trường hợp có thể xảy ra trong thực tế khi người dùng viết.
* Liên tục cập nhật các quy chuẩn tiếng Việt để có thể chuẩn hóa chính xác.
* Mở rộng data về các từ thông dụng và tiếng lóng để có thể xử lý câu hiệu quả.
* Cuối cùng là cải thiện mô hình huấn luyện bằng các thông số tối ưu với dataset. Từ đó giúp nâng cao được tỉ lệ của mô hình.

TÀI LIỆU THAM KHẢO

1. <https://nguyenvanhieu.vn/phan-loai-van-ban-tieng-viet>
2. <https://viblo.asia/p/phan-tich-phan-hoi-khach-hang-hieu-qua-voi-machine-learningvietnamese-sentiment-analysis-Eb85opXOK2G>
3. <https://github.com/huynhminhtan/thuattoanthongminh>
4. <https://nguyenvanhieu.vn/cach-crawl-du-lieu-web-bang-python/#cac-thu-vien-crawl-du-lieu-trong-python>
5. <https://telehub.vn/tinh-nang-tong-dai/sentiment-analysis>
6. <https://hocjavascript.net/angular/angular-la-gi-uu-diem-va-nhuoc-diem>
7. <https://ichi.pro/vi/phan-tich-cam-xuc-so-sanh-3-cach-tiep-can-pho-bien-naive-bayes-lstm-va-vader-188386384236425>
8. [www.noron.vn/post/gioi-thieu-ve-support-vector-machine-trong-machine-learning-40dxtjcmrdye](http://www.noron.vn/post/gioi-thieu-ve-support-vector-machine-trong-machine-learning-40dxtjcmrdye)
9. <http://maitrongnghia.com/2020/04/logistic-regression>
10. https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj
11. <https://www.academia.edu/25313121/X%C3%82Y_D%E1%BB%B0NG_H%E1%BB%86_TH%E1%BB%90NG_PH%C3%82N_LO%E1%BA%A0I_T%C3%80I_LI%E1%BB%86U_TI%E1%BA%BENG_VI%E1%BB%86T>
12. <https://dominhhai.github.io/vi/2017/10/what-is-lstm/>
13. <https://github.com/huynhminhtan/thuattoanthongminh>
14. <https://github.com/binhvq/news-corpus#%C4%91%E1%BB%8Bnh-d%E1%BA%A1ng-mongodb-dump>
15. <https://www.youtube.com/watch?v=EawbYWaTP_k>
16. <https://www.analyticsvidhya.com/blog/2021/06/natural-language-processing-sentiment-analysis-using-lstm/>
17. <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/>
18. https://www.edureka.co/blog/classification-in-machine-learning/#log