**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**

**VÕ KIỀU HOA – 18520767**

**NGUYỄN XUÂN VINH – 18521655**

**ĐỒ ÁN MÔN HỌC**

**MÔN: HỌC MÁY THỐNG KÊ**

**LỚP: DS102.K21**

# PHÂN TÍCH CUNG BẬC CẢM XÚC CỦA CÁC BÌNH LUẬN TIẾNG VIỆT TRÊN MẠNG XÃ HỘI

## BỘ DỮ LIỆU: UIT – VSMEC

**Analyze the emotional type of Vietnamese comments on social network**

**SINH VIÊN LỚP DS103. K21 – HỌC MÁY THỐNG KÊ**

**TP. HỒ CHÍ MINH, 2020**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**

**VÕ KIỀU HOA – 18520767**

**NGUYỄN XUÂN VINH – 18521655**

**ĐỒ ÁN MÔN HỌC**

**MÔN: HỌC MÁY THỐNG KÊ**

**LỚP: DS102.K21**

# PHÂN TÍCH CUNG BẬC CẢM XÚC CỦA CÁC BÌNH LUẬN TIẾNG VIỆT TRÊN MẠNG XÃ HỘI

## BỘ DỮ LIỆU: UIT – VSMEC

**Analyze the emotional type of Vietnamese comments on social network**

**SINH VIÊN LỚP DS103. K21 – HỌC MÁY THỐNG KÊ**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS NGUYỄN TẤN TRẦN MINH KHANG**

**Th.S VÕ DUY NGUYÊN**

**TP. HỒ CHÍ MINH, 2020**

# MỤC LỤC

# DANH MỤC HÌNH

# DANH MỤC BẢNG

# DANH MỤC TỪ VIẾT TẮT

UIT - VMEC: University of Information Technology - Vietnamese Social Media Emotion Corpus

MultinomialNB: Multinomial Naïve Bayes

SVM: Support Vector Machine

Bi – LSTM: Bi-directional Long Short-Term Memory

LSTM: Long Short-Term Memory

RNN: Recurrent Neural Network

# TÓM TẮT ĐỒ ÁN

Ngày nay, mạng xã hội trở thành một trong những thứ không thể thiếu của con người trong cuộc sống. Việc tương tác trên mạng xã hội trở nên phổ biến hơn. Từ đó hình thành nhiều vấn đề đáng quan tâm, chẳng hạn như việc phân tích cảm xúc trong bình luận. Nhóm em với mong muốn tìm tòi, học hỏi thêm nhiều thứ từ vấn đề trên nên chúng em đã chọn đề tài này.

Đồ án nhằm khái quát về mô hình phân tích cung bậc cảm xúc của các bình luận tiếng Việt trên mạng xã hội với bộ dữ liệu UIT – VSMEC. Mô hình sử dụng ngôn ngữ lập trình Python kết hợp ứng dụng các phương pháp, thuật toán trong máy học. Trong đồ án này, kết quả không đưa ra dưới dạng 0, 1 mà ở mức độ chi tiết hơn. Cụ thể, kết quả phân tích được biểu hiện với nhiều cảm xúc hơn: thích thú, buồn bã, tức giận, ghê tởm, sợ hãi, bất ngờ và cảm xúc khác.

Đồ án gồm những nội dung chính sau:

- Chương 1: Giới thiệu và phân tích đặc điểm bộ dữ liệu
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phân tích mô hình
- Chương 4: Kết quả thực nghiệm
- Chương 5: Kết luận

# Chương 1. GIỚI THIỆU VÀ PHÂN TÍCH ĐẶC ĐIỂM CỦA BỘ DỮ LIỆU

## 1.1 Giới thiệu về bộ dữ liệu

### *1.1.1 Tên bộ dữ liệu: UIT – VSMEC*

### *1.1.2 Nguồn*

- Google Drive: https://drive.google.com/drive/folders/1HooABJyrddVGzll7fgkJ6VzkG_XuWfRu
- Bộ dữ liệu được thu thập bởi: Vọng Anh Hồ, Dương Huỳnh-Công Nguyên, Danh Hoàng Nguyên, Linh Thị-Văn Phạm, Đức-Vũ Nguyễn, Kiệt Văn Nguyễn, và Ngân Lưu-Thúy Nguyễn. [1]
- Bộ dữ liệu được thu thập từ mạng xã hội Facebook bằng cách sử dụng API Facebook để lấy bình luận tiếng Việt từ các bài đăng công khai.

### *1.1.3 Nội dung bộ dữ liệu*

- Bộ dữ liệu dùng để phân tích cung bậc cảm xúc của các bình luận của người dùng trên mạng xã hội.
- Bộ dữ liệu gồm 3 tập tin: Tập huấn luyện (train_nor_8121.xlsx), tập kiểm thử (test_nor_811.xlsx) , tập kiểm định (valid_nor_811.xlsx)

### *1.1.4 Bài toán*

- Phân loại cảm xúc của các bình luận trên mạng xã hội
- Input: Sentence (câu bình luận)
- Output: Đưa ra một nhãn trong các nhãn sau: Enjoyment, Sadness, Anger, Disgust, Fear, Surprise, Other (Nhãn thể hiện cảm xúc của câu bình luận)

## 1.2 Phân tích đặc điểm của bộ dữ liệu

### *1.2.1 Mô tả chi tiết bộ dữ liệu*

#### *1.2.1.1 Kích thước bộ dữ liệu và số thuộc tính*

- Bộ dữ liệu gồm 6927 điểm dữ liệu được chia làm 3 tập: Tập huấn luyện (train_nor_8121.xlsx – 5548 điểm dữ liệu – 80% bộ dữ liệu), tập kiểm thử (test_nor_811.xlsx – 693 điểm dữ liệu – 10% bộ dữ liệu) , tập kiểm định (valid_nor_811.xlsx – 686 điểm dữ liệu – 10% bộ dữ liệu).
- Mỗi điểm dữ liệu gồm 2 thuộc tính: Sentence, Emotion.
- Hình minh họa dữ liệu trong bộ dữ liệu

| | Emotion | Sentence |
|---|---|---|
| **713** | Sadness | người ta có bạn bè nhìn vui thật |
| **1827** | Surprise | cho nghỉ viêc mói đúng sao goi là kỷ luật |
| **1166** | Disgust | kinh vãi 😟 |
| **228** | Fear | nhà thì không xa lắm nhưng chưa bao giờ đi vì sợ bị đè bẹp luôn người =))))) |
| **1942** | Anger | bố không thích nộp đấy mày thích ý kiến không |
| **285** | Other | bắt cóc bỏ đĩa . quanh đi quẩn lại chỉ khổ dân đen thôi ! |
| **260** | Other | per khi nào bạn có bằng xe ôtô thì bạn mới hiểu . sai thì đã có ca |
| **115** | Enjoyment | một hành động đẹp đầy tính nhân văn ☺☺ |
| **1054** | Other | anh nhỏ nhẹ xin 100k rồi trước khi đi còn chào người đẹp , đi cẩn thận " |
| **575** | Other | per trình mày vẫn còn thấp chán =)) |
| **40** | Other | mấy ai được như vậy ?? |
| **290** | Enjoyment | có ai như tao vô đọc bình luận  tao cười tao xỉu |
| **1798** | Disgust | ủa má . 2 đứa yêu nhau vô duyên và sân si giống nhau vậy ? |
| **23** | Enjoyment | góp vui cho chủ thớt 😂😂 |
| **1044** | Sadness | sống 17 năm chưa crush ai bao giờ |
| **284** | Other | ổng quay thiệt mà sao chửi ổng thế . không quay sao máy bạn xem được đó :)) |
| **211** | Disgust | thần kinh cái đầu mày ... |
| **1180** | Enjoyment | nghe ngọt thế :(( |
| **491** | Other | sao lại cứ phải thế nhỉ |
| **1467** | Enjoyment | mệt vãi .. mà cười cũng vãi ?????? |
| **384** | Fear | kinh khủng thật :(( |
| **12** | Other | hậu duệ mặt trời đấy chứ đùa đâu nhá 😳😳 |
| **1814** | Enjoyment | tôi ở phú yên và anh này nói rất chuẩn < 3 |
| **1027** | Enjoyment | đứa nào bị táo bón thâm niên cứ ra cầu này mà nhảy đảm bảo hiệu quả ngay lần đầu trải nghiệm  |
| **1953** | Surprise | trơi ơi sao lại cháy chứ |
| **235** | Disgust | có những người không có ý thức bỏ quần áo rách hoặc mặc rồi để vào tủ để đỡ phải đi vứt ... |
| **795** | Sadness | per khi ế nhưng vẫn phải giương mắt nhìn bọn nó ân ái |

*Hình 1. Hình minh họa dữ liệu trong tập Test của bộ dữ liệu*

#### *1.2.1.2 Mô tả chi tiết các thuộc tính*

| Số thứ tự | Tên thuộc tính | Mô tả thuộc tính | Kiểu dữ liệu | Miền giá trị |
|---|---|---|---|---|
| 1 | Sentence | Câu bình luận của người dùng mạng xã hội | String | Chữ cái, chữ số, kí tự đặc biệt. |
| 2 | Emotion | Nhãn phân loại các cảm xúc | String | Enjoyment, Sadness, Surprise, Disgust, Fear, Anger, Other. |

*Bảng 1. Bảng mô tả chi tiết các thuộc tính trong bộ dữ liệu*

#### *1.2.1.3 Mô tả chi tiết các nhãn*

| Số thứ tự | Tên nhãn | Ý nghĩa |
|---|---|---|
| 1 | Anger | Nhãn cho những bình luận thể hiện sự giận dữ, cảm giác phiền toái, bực tức, tức giận,… |
| 2 | Disgust | Nhãn cho những bình luận thể hiện sự ghê tởm, không thích, ác cảm,… |
| 3 | Enjoyment | Nhãn cho những bình luận thể hiện cảm xúc thích thú, vui vẻ, nó bao gồm cả các trạng thái tích cực, hài lòng, lạc quan hoặc những cảm xúc tương tự như vậy. |
| 4 | Fear | Nhãn cho những bình luận thể hiện sự lo lắng, sợ hãi, có thể hơn thế nữa là hoảng loạn, kinh hoang – hỗn hợp của sợ hãi, ghê tởm, sốc. |

| | | |
|---|---|---|
| 5 | Other | Nhãn cho những bình luận không thuộc các loại cảm xúc: Anger, Disgust, Enjoyment, Fear, Other, Sadness, Surprise, hoặc không chứa đựng cảm xúc. |
| 6 | Sadness | Nhãn cho những bình luận thể hiện cảm xúc buồn, chán nản, thất vọng, tuyệt vọng, đau khổ, thống khổ, hoặc những cảm xúc tương tự như vậy. |
| 7 | Surprise | Nhãn cho những bình luận thể hiện sự bất ngờ, như vừa gặp phải một điều gì đó khó tin, mạnh hơn có thể là sốc. |

*Bảng 2. Bảng mô tả chi tiết các nhãn trong bộ dữ liệu*

#### *1.2.1.4 Ví dụ về phân loại cảm xúc câu*

| Số thứ tự | Sentence | Emotion |
|---|---|---|
| 1 | hãy xử tù chung thân bọn người mất nhân tính gây ra nghiệp chướng này | Anger |
| 2 | ngửa thấy mùi là đã thấy kinh rồi | Disgust |
| 3 | ngưỡng mộ quá đi ạ . chúc 2 ac mãi mãi hạnh phúc như này nha :))) | Enjoyment |
| 4 | xã hội thật phức tợp , huhu . không muốn sống chung loài người nữa , tao về núi đây | Fear |
| 5 | có gì vui không nhỉ | Other |
| 6 | còn buồn hơn là cố quan tâm thế nào họ vẫn không hề rep lại | Sadness |
| 7 | hot đến vậy cơ à ? mình chưa nghe bao giờ luôn | Surprise |

*Bảng 3. Bảng ví dụ về các câu bình luận và nhãn tương ứng*

### *1.2.2 Thống kê bộ dữ liệu*

#### *1.2.2.1 Thống kê dữ liệu tập huấn luyện (Train set)*

| Số thứ tự | Tên nhãn | Số lượng | Tỉ lệ |
|---|---|---|---|
| 1 | Anger | 391 | 0.070 |
| 2 | Disgust | 1071 | 0.193 |
| 3 | Enjoyment | 1558 | 0.281 |
| 4 | Fear | 318 | 0.057 |
| 5 | Other | 1021 | 0.184 |
| 6 | Sadness | 947 | 0.171 |
| 7 | Surprise | 242 | 0.044 |

*Bảng 4. Thống kê dữ liệu tập huấn luyện*

#### *1.2.2.2 Thống kê dữ liệu tập kiểm thử (Test set)*

| Số thứ tự | Tên nhãn | Số lượng | Tỉ lệ |
|---|---|---|---|
| 1 | Anger | 40 | 0.058 |
| 2 | Disgust | 132 | 0.191 |
| 3 | Enjoyment | 193 | 0.279 |
| 4 | Fear | 46 | 0.066 |
| 5 | Other | 129 | 0.186 |
| 6 | Sadness | 116 | 0.167 |
| 7 | Surprise | 37 | 0.053 |

*Bảng 5. Thống kê dữ liệu tập kiểm thử*

#### *1.2.2.3 Thống kê dữ liệu tập kiểm định (Validation set)*

| Số thứ tự | Tên nhãn | Số lượng | Tỉ lệ |
|---|---|---|---|
| 1 | Anger | 49 | 0.071 |
| 2 | Disgust | 135 | 0.197 |
| 3 | Enjoyment | 214 | 0.312 |
| 4 | Fear | 31 | 0.045 |
| 5 | Other | 141 | 0.206 |
| 6 | Sadness | 86 | 0.125 |
| 7 | Surprise | 30 | 0.044 |

*Bảng 6. Thống kê dữ liệu tập kiểm định*

# Chương 2. CƠ SỞ LÝ THUYẾT

## 2.1 Học máy (Machine Learning)

### *2.1.1 Multinomial Naive Bayes (MultinomialNB) [2]*

Mô hình MultinomialNB được sử dụng chủ yếu trong các bài toán phân loại văn bản mà vector đặc trưng được xây dựng trên thuật toán bag of words (BoW). Ở mô hình, các vector đặc trưng là các giá trị số tự nhiên mà giá trị thể hiện số lần xuất hiện trong văn bản. Ta tính xác suất các từ xuất hiện trong văn bản P($x_i|y$) như sau:

$$\mathbf{P(x_i|y) = \frac{N_i}{N_c}}$$

Trong đó:

- $N_i$ *là tổng số lần từ* $x_i$ *không xuất hiện lần nào trong văn bản.*
- $N_c$ *là tổng số lần từ của tất cả các từ* $x_1, \dots x_n$ *xuất hiện trong văn bản.*

Công thức trên có hạn chế là khi từ $x_i$ không xuất hiện lần nào trong văn bản, ta sẽ có $N_i$ = 0. Điều nãy sẽ làm cho P($x_i|y$) = 0.

Để khắc phục vấn đề này, người ta sử dụng kĩ thuật gọi là Laplace Smoothing bằng cách cộng thêm vào tử và mẫu để giá trị luôn khác 0.

$$\mathbf{P(x_i|y) = \frac{N_i + \alpha}{N_c + d\alpha}}$$

Trong đó:

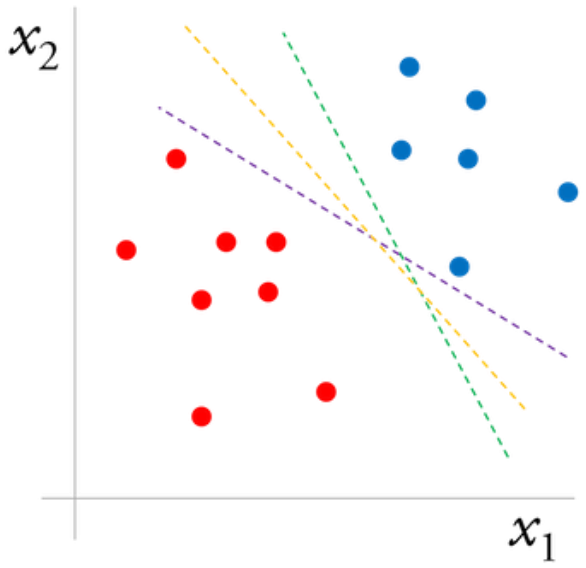
- *α thường là số dương, bằng 1.*
- *dα được cộng vào mẫu để đảm bảo.*

### *2.1.2 Support Vector Machine (SVM) [3]*

#### *2.1.2.1 Định nghĩa*

Support Vector Machine (SVM) là một thuật toán thuộc nhóm Supervised Learning (Học có giám sát) dùng để phân chia dữ liệu (Classification) thành các nhóm riêng biệt.

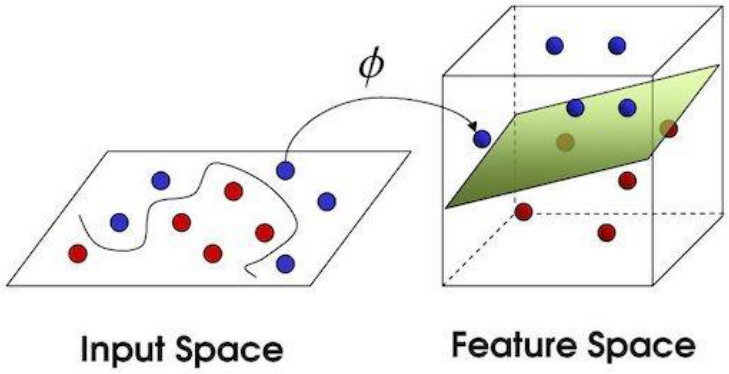
Hình dung ta có bộ data gồm các điểm xanh và đỏ đặt trên cùng một mặt phẳng. Ta có thể tìm được đường thẳng để phân chia riêng biệt các bộ điểm xanh và đỏ như hình bên dưới.




*Hình 2. Minh họa đường thẳng phân chia dữ liệu trong thuật toán SVM*

Với những bộ dữ liệu phức tạp hơn ta cần dùng thuật toán để ánh xạ bộ dữ liệu đó vào không gian nhiều chiều hơn (n chiều), từ đó tìm ra siêu mặt phẳng (hyperplane) để phân chia.

Ví dụ trong hình bên dưới là việc ánh xạ tập dữ liệu từ không gian 2 chiều sang không gian 3 chiều.

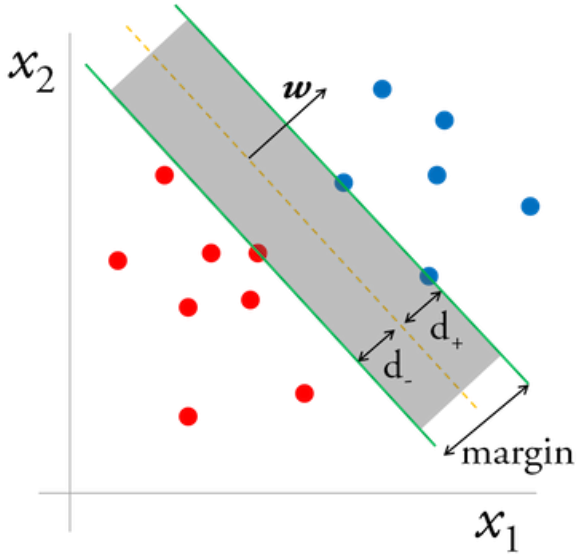



*Hình 3. Minh họa việc ánh xạ tập dữ liệu từ không gian 2 chiều sang 3 chiều*

#### *2.1.2.2 Tối ưu trong thuật toán SVM*

Các đường thẳng tối ưu là đường tạo cho ta cảm giác 2 lớp dữ liệu nằm cách xa nhau và cách xa đường đó nhất. Trong SVM, người ta sử dụng thuật ngữ là Margin.

Margin là khoảng cách giữa siêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp.




*Hình 4. Minh họa Margin trên không gian 2 chiều*

SVM tối ưu thuật toán bằng cách tìm maximize của margin này, từ đó tìm ra siêu phẳng **tốt nhất** để phân lớp dữ liệu.

Support Vectors là các điểm nằm trên 2 đường biên và chúng có nhiệm vụ hỗ trợ tìm ra siêu phẳng.

Mô hình dự đoán kết quả đầu ra của của những điểm dữ liệu mới dựa trên các vector đặc biệt này.

## 2.2 Học sâu (Deep Learning)

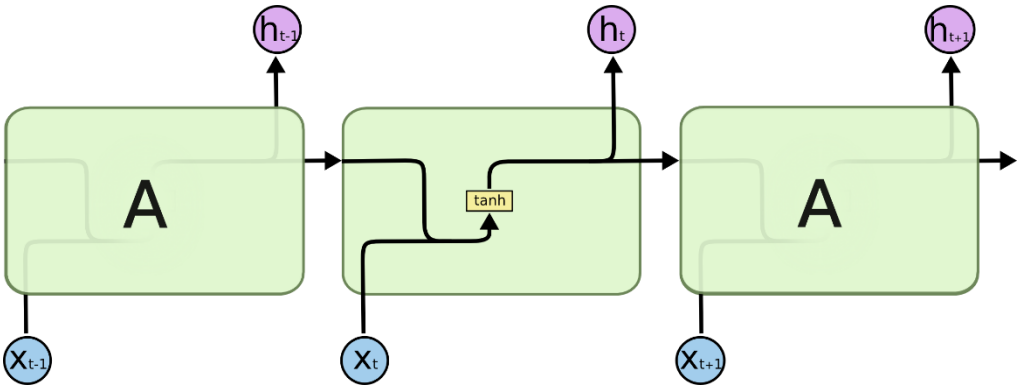
### *2.2.1 Long Short – Term Memory (LSTM) [4]*

#### *2.2.1.1 Giới thiệu Long Short – Term Memory (LSTM)*

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi *Hochreiter & Schmidhuber (1997)*, và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

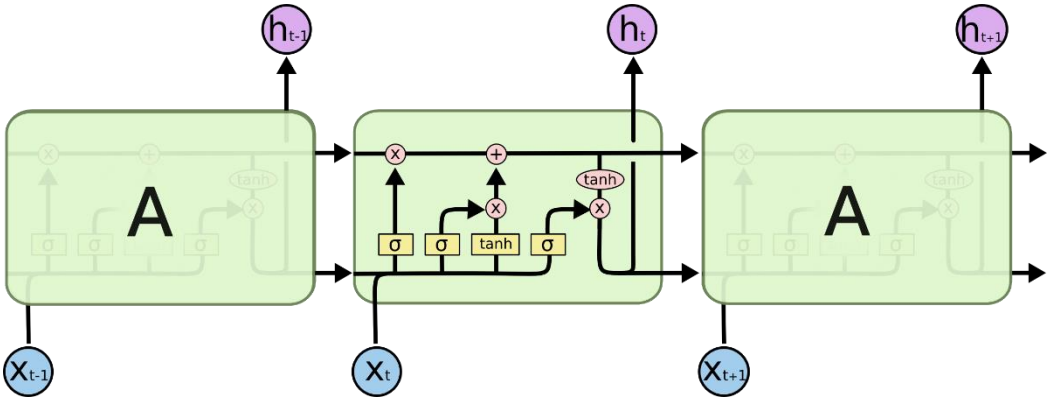
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng *tanh*.

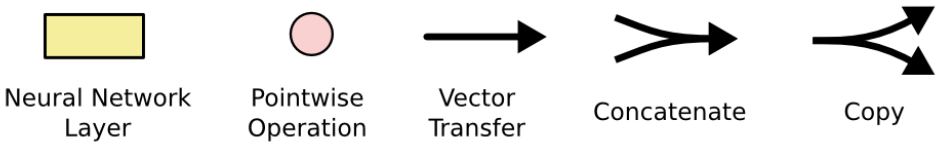



*Hình 5.The repeating module in a standard RNN contains a single layer.*

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.




*Hình 6. The repeating module in an LSTM contains four interacting layers.*

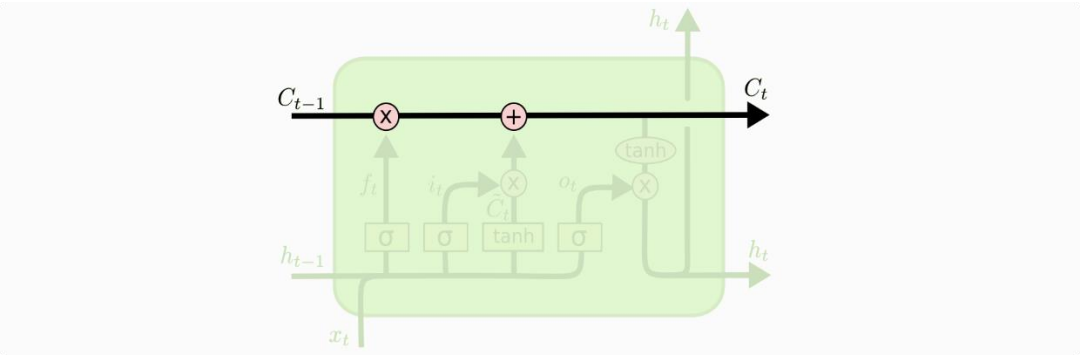



*Hình 7. Các ký hiệu trong cấu trúc LSTM*

#### *2.2.1.2 Ý tưởng cốt lõi của Long Short – Term Memory (LSTM)*

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

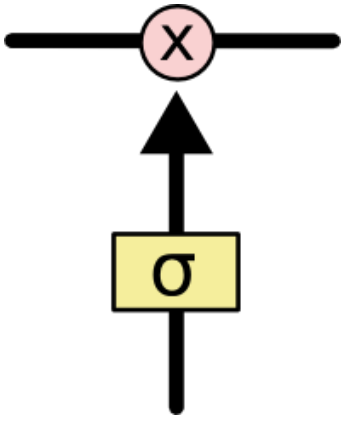
Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.




*Hình 8.Mô phỏng cấu trúc trạng thái tế bào của Long Short - Term Memory*

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



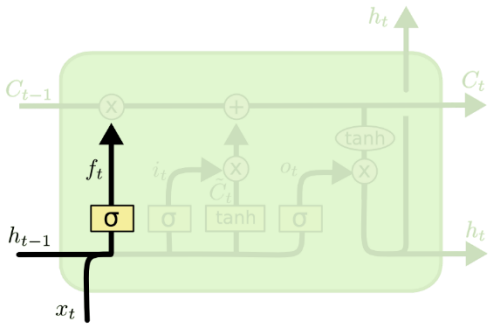

*Hình 9. Mô phỏng cổng*

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1][0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 00 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

#### *2.2.1.3 Bên trong Long Short – Term Memory (LSTM)*

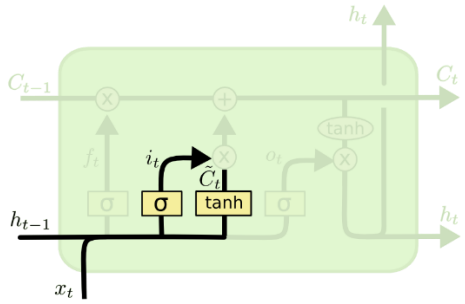
Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là $h_{t-1}$ và $x_t$ rồi đưa ra kết quả là một số trong khoảng [0, 1] cho mỗi số trong trạng thái tế bào $C_{t-1}$.Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.




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

*Hình 10. Mô phỏng Forget Gate trong LSTM*

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là "tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng *tanh* tạo ra một véc-tơ cho giá trị mới $\tilde{C}_t$nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

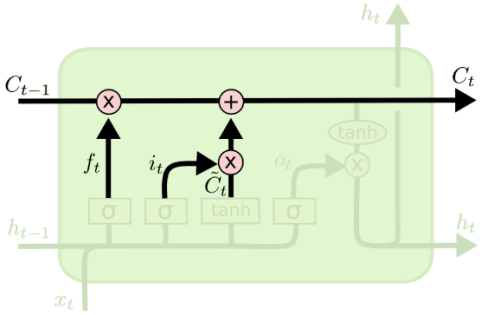



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

*Hình 11. Mô phỏng Input Gate trong LSTM (1)*

Giờ là lúc cập nhập trạng thái tế bào cũ $C_{t-1}$ thành trạng thái mới $C_t$. Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong.

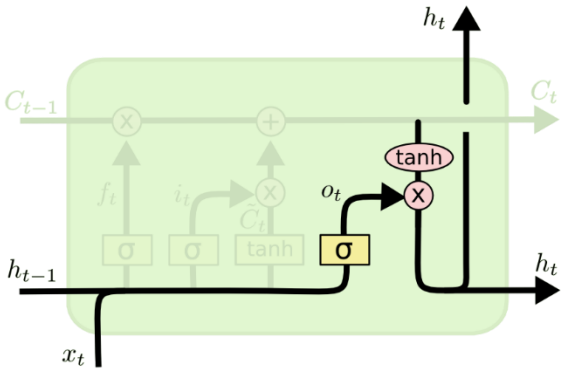
Ta sẽ nhân trạng thái cũ với $f_t$ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm $i_t * \tilde{C}_t$. Trạng thái mơi thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.




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

*Hình 12. Mô phỏng Input Gate trong LSTM (2)*

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 co giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.

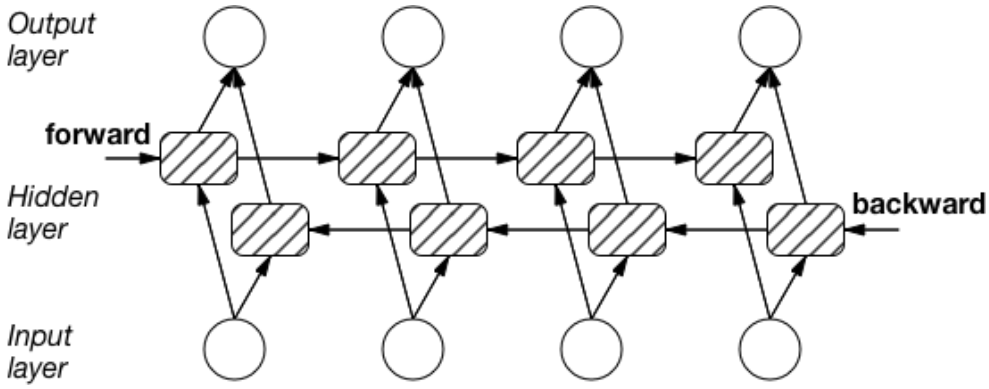



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

*Hình 13. Mô phỏng Output Gate trong LSTM*

### *2.2.2 Bi – Directional Long Short – Term Memory (Bi - LSTM) [5]*

Bi – Directional Long Short – Term Memory (Bi - LSTM) là một dạng mở rộng của Long Short – Term Memory (LSTM). Một kiến trúc Bi-LSTM thường chứa 2 mạng LSTM đơn được sử dụng đồng thời và độc lập để mô hình hoá chuỗi đầu vào theo 2 hướng: từ trái sang phải (forward LSTM) và từ phải sang trái (backward LSTM)




*Hình 14. Bidirectional LSTM = forward LSTM + backward LSTM*

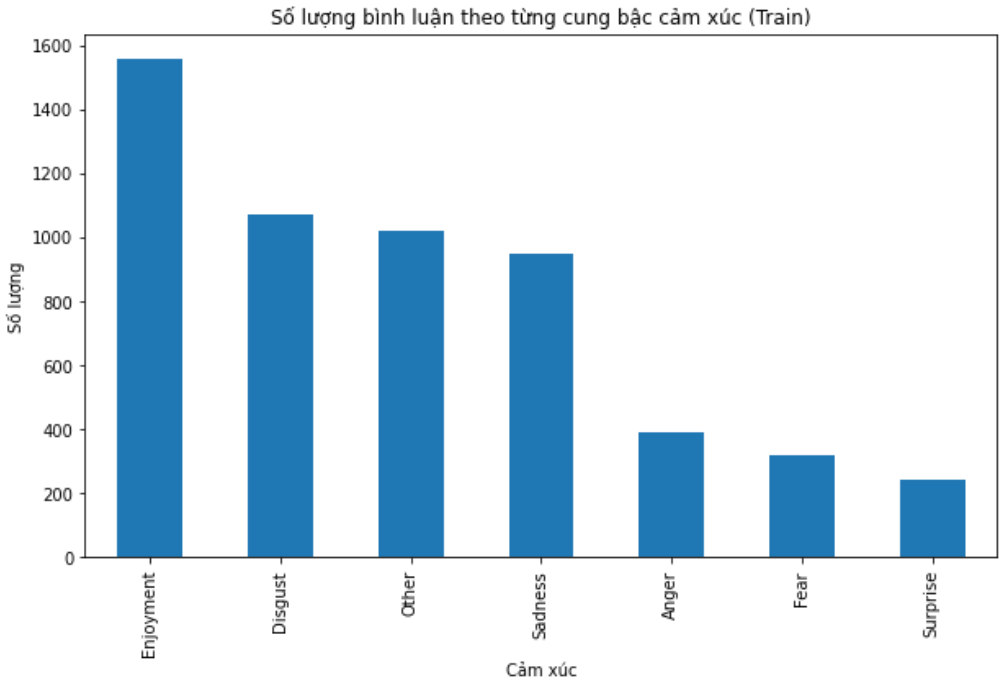
# Chương 3. PHÂN TÍCH MÔ HÌNH

## 3.1 Mô hình học máy MultinomialNB và SVM

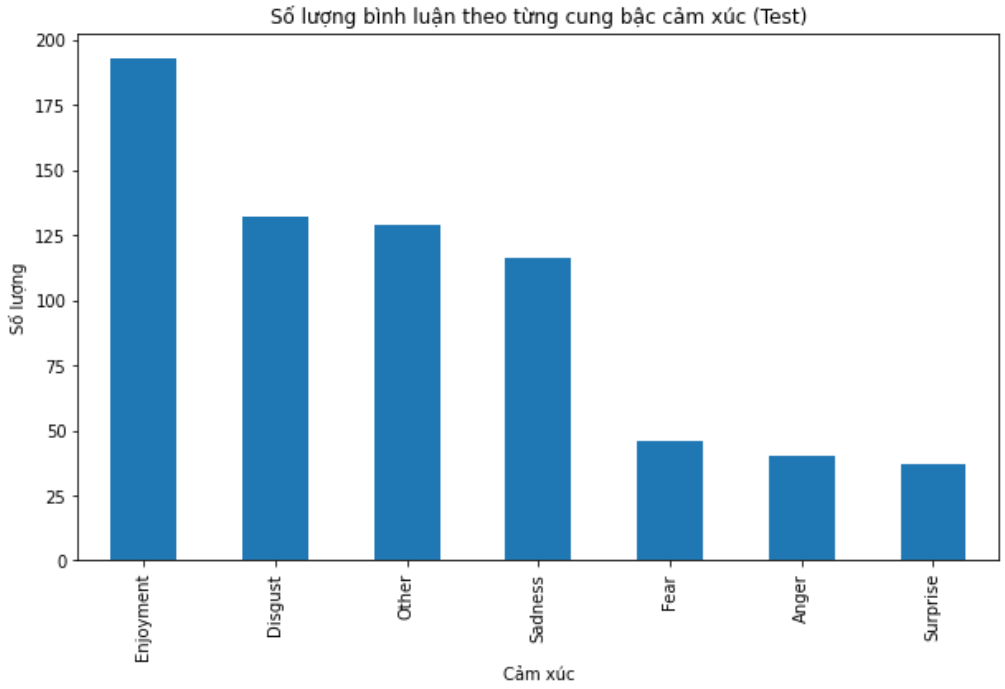
### *3.1.1 Khái quát về mô hình MultinomialNB và SVM*

Mô hình phân loại cung bậc cảm xúc của bình luận tiếng việt trên mạng xã hội bằng MultinomialNB và SVM.

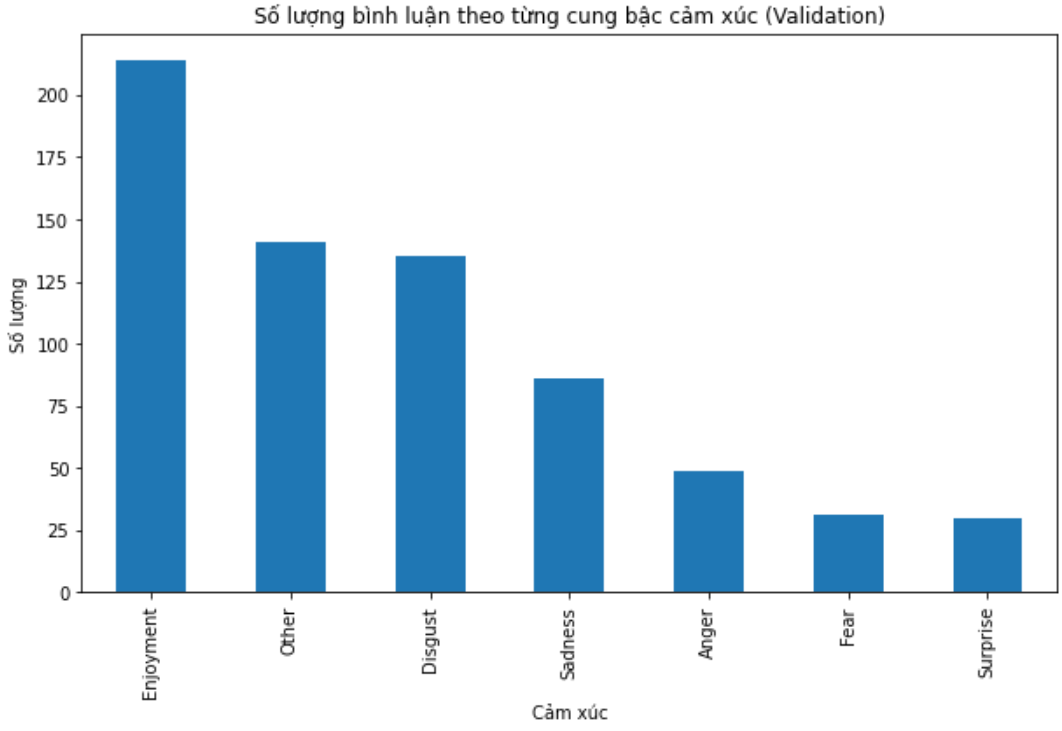
Sự phân bố các cung bậc cảm xúc được thể hiện ở biểu đồ sau:




*Hình 15. Biểu đồ thể hiện số lượng bình luận thuộc mỗi cung bậc cảm xúc trên tập Train.*




*Hình 16. Biểu đồ thể hiện số lượng bình luận thuộc mỗi cung bậc cảm xúc trên tập Test*




*Hình 17. Biểu đồ thể hiện số lượng bình luận thuộc mỗi cung bậc cảm xúc trên tập Validation*

Tập dữ liệu train_nor_811.xlsx dùng để huấn luyện dữ liệu. Tập dữ liệu valid_nor_811.xlsx dùng để kiểm tra trong quá trình huấn luyện và tập dữ liệu test_nor_811.xlsx dùng để đánh giá mô hình.

### *3.1.2 Chi tiết tiến trình mô hình MultinomialNB và SVM*

Bước đầu tiền xử lý, dữ liệu text của các bình luận tiếng việt sẽ được xử lý bằng *Countvectorizer* [6] và *TfidfTransformer* [7] từ mô hình *sklearn.feature_extraction.text* [8]. Nhiệm vụ của *Countvectorizer* là chuyển dữ liệu từ dạng text thành vector.

Nếu có một mảng các string corpus ta sẽ biến đổi mảng này thành cột vector có độ dài bằng số từ xuất hiện ít nhất một lần trong corpus. Giá trị của thành phần thứ i chính là số lần xuất hiện của từ đó trong string.

*TfidfTransformer* sẽ có nhiệm vụ chuẩn hoá vector được tạo ra từ *Countvectorizer* về dạng *Tf-idf*.

Sau khi hoàn thành tiền xử lý ta sẽ bắt đầu huấn luyện mô hình. Ở đây, ta có 2 mô hình để huấn luyện lần lượt là *MultinomialNB* và *SVM*.

Với mô hình *MultinomialNB* ta import từ mô hình *sklearn.naive_bayes* [9]: `from sklearn.naive_bayes import MultinomialNB`, sau đó, ta bắt đầu huấn luyện.

Còn với mô hình SVM thì ta import từ *sklearn.svm* [10]: `from sklearn.svm import SVC`, rồi bắt đầu huấn luyện.

## 3.2 Mô hình học sâu Bi-LSTM

### *3.2.1 Khái quát về mô hình Bi-LSTM*

Với mô hình này ta sử dụng một mô hình *pretrained* của *Fasttext* [11] thuộc dạng *Word Embedding* [12] trong tiếng Việt kết hợp với mạng *Bi-LSTM*.

*Word Embedding* là tên gọi chung của các mô hình ngôn ngữ và các phương pháp học theo đặc trưng trong xử lý ngôn ngữ tự nhiên (NLP), ở đó các từ hoặc cụm từ được ánh xạ sang các vector số (thường là số thực).

Đây là một công cụ đóng vai trò quan trọng đối với hầu hết các thuật toán, kiến trúc Machine Learning, Deep Learning trong việc xử lý input ở dạng text, do chúng chỉ có thể hiểu được input ở dạng là số, từ đó mới thực hiện các công việc phân loại, hồi quy,… Cụ thể trong mô hình này ta sử dụng *Word2Vec*.

### *3.1.3 Chi tiết tiến trình mô hình Bi-LSTM*

Tải mô hình pretrained. Vì các lý do chủ quan và khách quan mà nhóm em không thể tải mô hình pretrained về máy được. Nên đã chuyển sang phương án sử dụng Google Colab của Google để thực hiện. Sau khi tải hoàn tất mô hình pretrained trên Google Colab thì bắt đầu giải nén file.

Trích xuất đặc trưng của dữ liệu bằng cách gọi *text.Tokenizer* và tokenizer.fit_on_texts.

Bước tiếp theo, các bình luận tiếng Việt sẽ được vector hoá bằng phương thức *text_to_sequences* thuộc lớp đối tượng *Tokenizer*.

Tạo *embedding_index* được sử dụng như từ điển tính toán *embedding_matrix* mà sau đó sẽ được load vào lớp *Embedding* mặc định trong *Bi-LSTM* để phân tích dữ liệu mô hình *pretrained*.

Kế tiếp, tiến hành cài đặt mạng *Bi-LSTM* 1 chiều với số chiều dữ liệu đầu ra là 7 (do có 7 lớp cảm xúc Anger, Disgust, Enjoyment, Fear, Other, Sadness, Surprise) và hàm kích hoạt *sigmoid*.

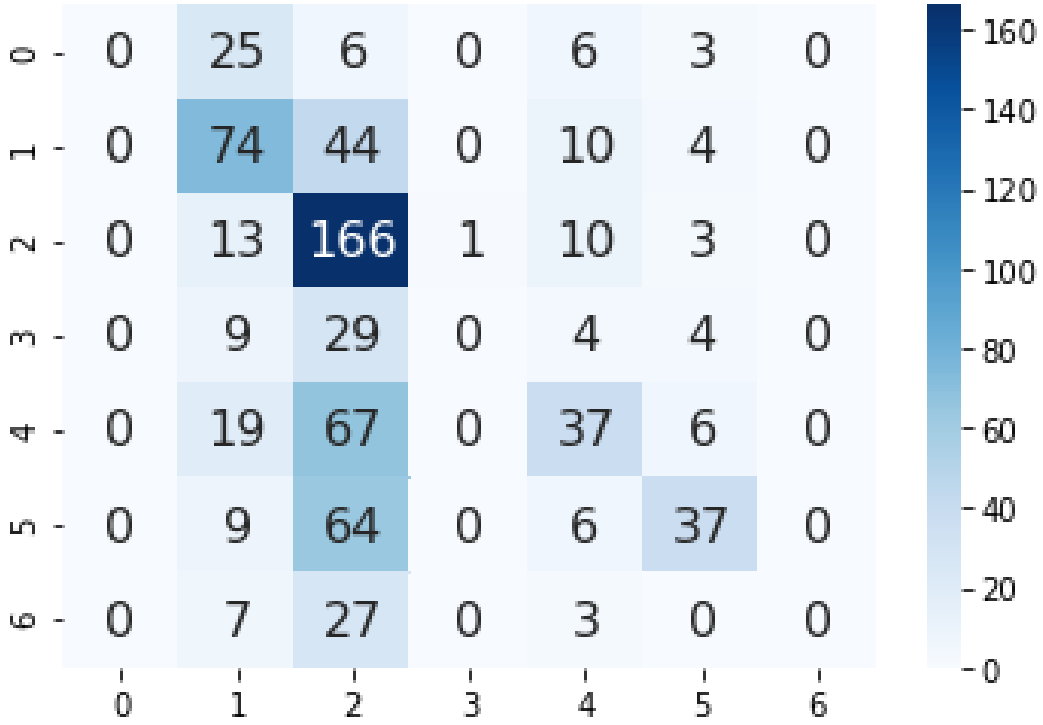
Bắt đầu huấn luyện trên tập train_nor_811.xlsx với số *epoch* bằng 5 và tập valid_nor_811.xlsx được lấy làm tập validation.

Cuối cùng đánh giá kết quả mô hình bằng tập test_nor_811.xlsx và lưu lại kết quả kiểm tra cùng với mô hình để tiện cho việc báo cáo.

# Chương 4. KẾT QUẢ THỰC NGHIỆM

## 4.1 Confusion matrix

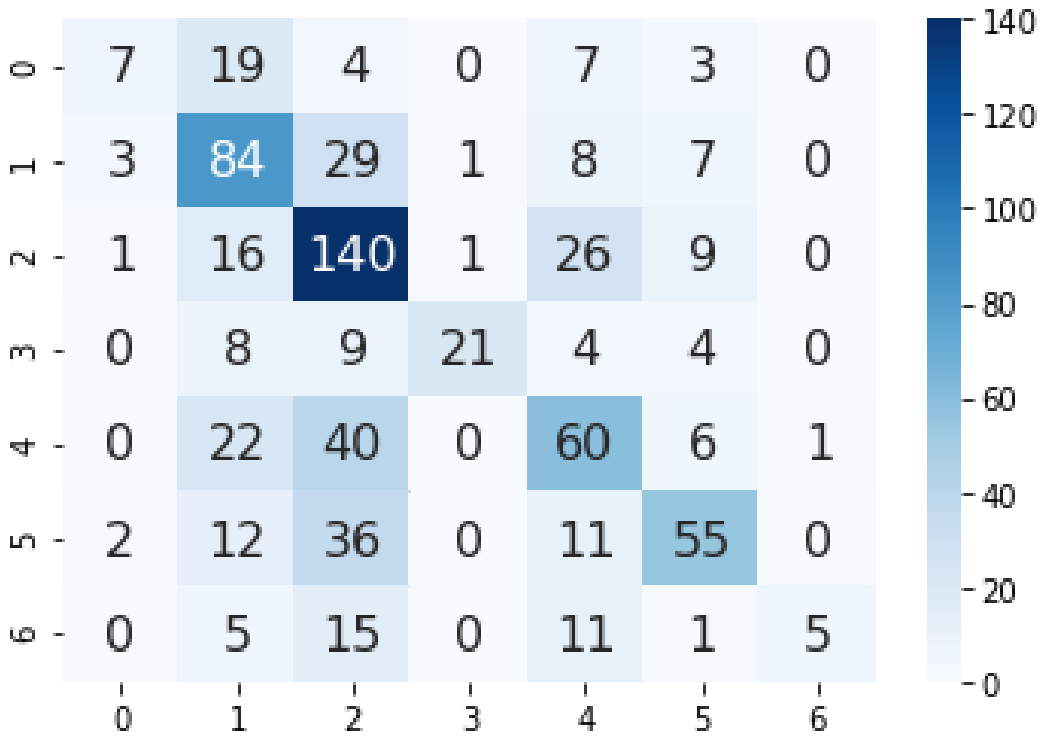
### *4.1.1 Confusion matrix trên tập Test*




*Hình 18. Confusion maxtrix bởi Naive Bayes với tập test*

➢ ***Nhận xét:***

- Điểm dữ liệu đúng = 0+74+166+0+37+37+0 = 314
- Điểm dữ liệu sai = 693 – 314 = 379
- Tổng điểm dữ liệu = 693
- Tỉ lệ phân loại điểm dữ liệu đúng = 314/693 = 0.4531 ~ 45.31%
- Tỉ lệ phân loại điểm dữ liệu sai = 1 -0.4531 = 0.5469 ~ 54.69%

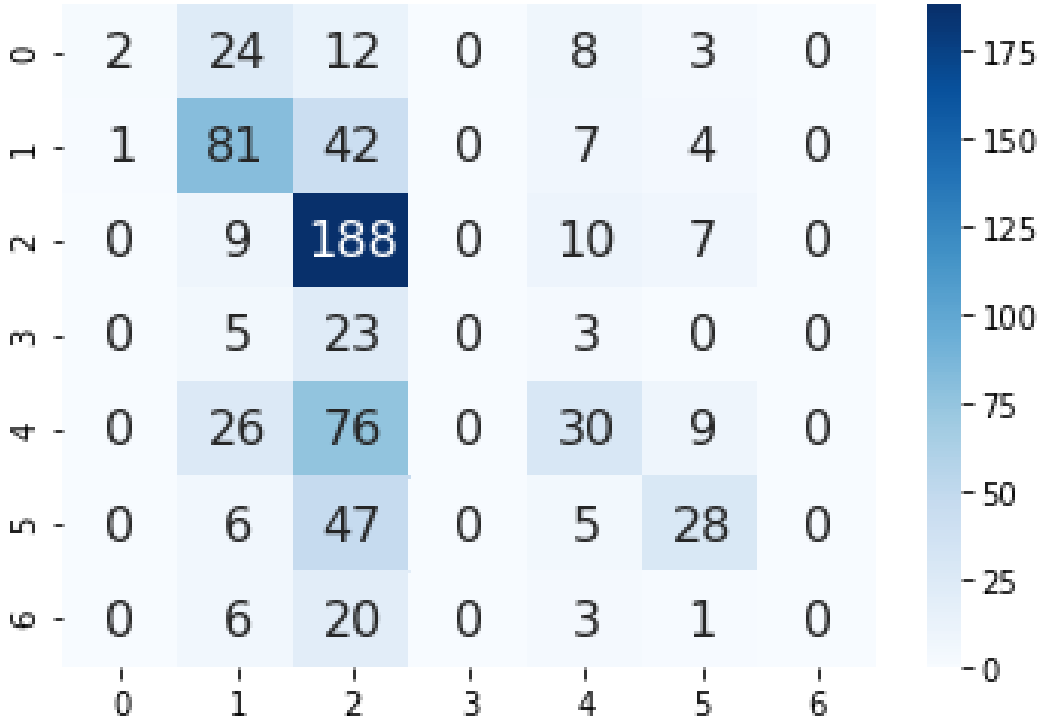



*Hình 19. Confusion maxtrix bởi SVM với tập test*

➢ ***Nhận xét:***

- Điểm dữ liệu đúng = 7+84+140+21+60+55+5 = 372
- Điểm dữ liệu sai = 693 – 372 = 321
- Tổng điểm dữ liệu = 693
- Tỉ lệ phân loại điểm dữ liệu đúng = 372/693 = 0.5368 ~ 53.68%
- Tỉ lệ phân loại điểm dữ liệu sai = 1 -0.5368 = 0.4632 ~ 46.32%

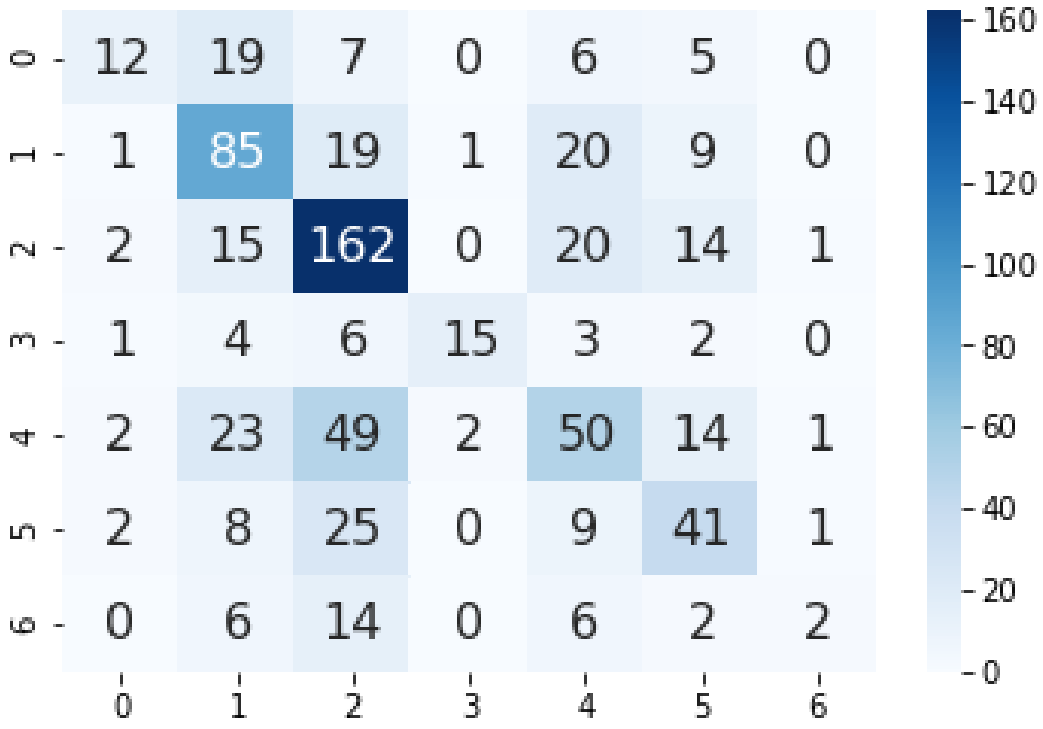
### *4.1.2 Confusion matrix trên tập Validation*




*Hình 20. Confusion maxtrix bởi Naive Bayes với tập validation*

➢ ***Nhận xét:***

- Điểm dữ liệu đúng = 2+81+188+0+30+28+0 = 329
- Điểm dữ liệu sai = 686 – 329 = 357
- Tổng điểm dữ liệu = 686
- Tỉ lệ phân loại điểm dữ liệu đúng = 329/686 = 0.4796 ~ 47.96%
- Tỉ lệ phân loại điểm dữ liệu sai = 1 -0.4796 = 0.5204 ~ 52.04%




*Hình 21. Confusion maxtrix bởi SVM với tập validation*

- ***Nhận xét:***
    - Điểm dữ liệu đúng = 12+85+162+15+50+41+2 = 367
    - Điểm dữ liệu sai = 686 – 367 = 319
    - Tổng điểm dữ liệu = 686
    - Tỉ lệ phân loại điểm dữ liệu đúng = 367/686 = 0.5350 ~ 53.50%
    - Tỉ lệ phân loại điểm dữ liệu sai = 1 – 0.5350 = 0.4650 ~ 46.50%

## 4.2 Accuracy, precision, recall, f1-score

### *4.2.1 Multinomial Naive bayes (MultinomialNB)*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **Anger** | 0.00 | 0.00 | 0.00 | 40 |
| **Disgust** | 0.47 | 0.56 | 0.51 | 132 |
| **Enjoyment** | 0.41 | 0.86 | 0.56 | 193 |
| **Fear** | 0.00 | 0.00 | 0.00 | 46 |
| **Other** | 0.49 | 0.29 | 0.36 | 129 |
| **Sadness** | 0.65 | 0.32 | 0.43 | 116 |
| **Surprise** | 0.00 | 0.00 | 0.00 | 37 |
| | | | | |
| **accuracy** | | | 0.45 | 693 |
| **macro avg** | 0.29 | 0.29 | 0.27 | 693 |
| **weighted avg** | 0.40 | 0.45 | 0.39 | 693 |

*Bảng 7. Các độ đo trên tập Test với mô hinh MultinomialNB*

### *4.2.2 Support Vector Machine (SVM)*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **Anger** | 0.54 | 0.17 | 0.26 | 40 |
| **Disgust** | 0.51 | 0.64 | 0.56 | 132 |
| **Enjoyment** | 0.51 | 0.73 | 0.60 | 193 |
| **Fear** | 0.91 | 0.46 | 0.61 | 46 |
| **Other** | 0.47 | 0.47 | 0.47 | 129 |
| **Sadness** | 0.65 | 0.47 | 0.55 | 116 |
| **Surprise** | 0.83 | 0.14 | 0.23 | 37 |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **accuracy** | | | 0.54 | 693 |
| **macro avg** | 0.63 | 0.44 | 0.47 | 693 |
| **weighted avg** | 0.57 | 0.54 | 0.52 | 693 |

*Bảng 8. Các độ đo trên tập Test với mô hinh SVM*

### *4.2.3 Bi – Directional Long Short – Term Memory (Bi-LSTM)*

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **Anger** | 0.46 | 0.60 | 0.52 | 40 |
| **Disgust** | 0.57 | 0.50 | 0.53 | 132 |
| **Enjoyment** | 0.68 | 0.61 | 0.64 | 193 |
| **Fear** | 0.66 | 0.76 | 0.71 | 46 |
| **Other** | 0.45 | 0.47 | 0.46 | 129 |
| **Sadness** | 0.59 | 0.68 | 0.63 | 116 |
| **Surprise** | 0.50 | 0.49 | 0.49 | 37 |
| | | | | |
| **Accuracy** | | | 0.88 | 693 |
| **Micro avg** | 0.58 | 0.58 | 0.58 | 693 |
| **macro avg** | 0.56 | 0.59 | 0.57 | 693 |
| **weighted avg** | 0.58 | 0.58 | 0.58 | 693 |
| **Samples avg** | 0.58 | 0.58 | 0.58 | 693 |

*Bảng 9. Các độ đo trên tập Test với mô hinh Bi-LSTM*

### *4.2.4 Nhận xét kết quả thực nghiệm*

Dựa vào các bảng độ đo ở trên ta có bảng sau:

| | **precision** | **recall** | **f1-score** | **Accuracy** |
|---|---|---|---|---|
| **MultinomialNB** | 0.40 | 0.45 | 0.39 | 0.45 |

| | | | | |
|---|---|---|---|---|
| **SVM** | 0.57 | 0.54 | 0.52 | 0.54 |
| **Bi-LSTM** | 0.58 | 0.58 | 0.58 | 0.88 |

*Bảng 10. Kết quả đánh giá 3 mô hình MultinomialNB, SVM và Bi-LSTM*

Từ bảng 10, ta thấy mô hình *MultinomialNB* và mô hình *SVM* cùng sử dụng *Countvectorizer* và *TfidfTransformer* cho việc tiền xử lý dữ liệu. Nên ta so sánh chúng trước.

Đối với mô hình MultinomialNB, kết quả đánh giá trên tập test với precision, recall, f1-score, accuracy lần lượt là 40%, 45%, 39%, 45%. Kết quả không tốt cho lắm.

Tuy nhiên ở mô hình SVM chúng ta có kết quả khả quan hơn, với kết quả đánh giá trên tập test với precision, recall, f1-score, accuracy lần lượt là 57%, 54%, 52%, 54%.

Có thể nhận thấy cùng sử dụng Countvectorizer và TfidfTransformer để tiền xử lý dữ liệu nhưng mô hình SVM có kết quả tốt hơn so với mô hình MultinomialNB. Điều đó cho thấy mô hình SVM phù hợp với bộ dữ liệu này hơn mô hình MultinomialNB.

Mô hình Bi-LSTM áp dụng một mô hình pretrained để tạo vector từ văn bản nhưng có áp dụng cấu trúc ngữ nghĩa. liên kết từ và kết hợp sử dụng Bi-LSTM cho phép học sâu hơn.

Do vậy, với kết quả kiểm tra trên tập Test với precision, recall, f1-score, accuracy lần lượt là 58%, 58%, 58%, 88%. Cao hơn hẳn so với những mô hình học máy truyền thống như MultinomialNB hay SVM. Tuy nhiên, mô hình này có một nhược điểm, đó là thời gian huấn luyện sẽ lâu hơn rất nhiều so với mô hình học máy truyền thống.

## Chương 5. KẾT LUẬN

Trong đồ án môn học Học máy Thống kê này, chúng em đã giải quyết thành công bài toán phân tích các cung bậc cảm xúc của con người trong những bình luận trên mạng xã hội. Với bộ dữ liệu gồm 6927 câu bình luận được phân loại một trong bảy nhãn cảm xúc: Enjoyment, Sadness, Anger, Surprise, Fear, Disgust, Other.

Chúng em đã sử dụng 2 mô hình trong học máy (Machine Learning) và 1 mô hình trong học sâu (Deep Learning) để giải quyết bài toán đặt ra. Cụ thể f1-score của mô hình MultinomialNB đạt 0.39 và Accuracy đạt 0.45; f1-score của mô hình SVM đạt 0.52 và Accuracy đạt 0.54; f1-score của hô hình Bi-LSTM (Deep Learning) đạt 0.58 và Accuracy đạt 0.88. Qua đó, ta thấy sử dụng Deep Learning trong huấn luyện mô hình cho bộ dữ liệu này sẽ cho ra mô hình hoạt động tốt hơn.

Tuy nhiên, vẫn còn nhược điểm về thời gian huấn luyện mô hình. Trong tương lai, chúng em muốn khám phá và tìm kiếm thêm nhiều phương pháp ứng dụng học máy, học sâu khác tối ưu hơn để cải thiện chất lượng mô hình, đồng thời nhằm nâng cao tỉ lệ dự đoán chính xác của mô hình.

# TÀI LIỆU THAM KHẢO


[1] D. H. C. N. D. H. N. L. T. V. P. D. V. N. K. V. N. N. L. T. N. Vong Anh Ho, "Emotion Recognition for Vietnamese Social Media Text," 2019.

[2] N. Doan, "1 UP Note," [Online]. Available: https://1upnote.me/post/2018/11/ds-ml-naive-bayes/#2-m%C3%B4-h%C3%ACnh-multinomial.

[3] N. Doan, "1 UP Note," [Online]. Available: https://1upnote.me/post/2018/10/ds-ml-svm/#support-vectors.

[4] colah, "Understanding-LSTM," [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[5] L. V. Thang, "Nhận diện tên riêng trong văn bản với Bidirectional Long Short-Term Memory và Conditional Random Field," [Online]. Available: https://medium.com/@lngvietthang/nh%E1%BA%ADn-di%E1%BB%87n-t%C3%AAn-ri%C3%AAng-trong-v%C4%83n-b%E1%BA%A3n-v%E1%BB%9Bi-bidirectional-long-short-term-memory-v%C3%A0-conditional-random-b11bc75c512b.

[6] "CountVectorizer," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[7] "TfidfTransformer," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

[8] "Feature extraction," [Online]. Available: https://scikit-learn.org/stable/modules/feature_extraction.html.

[9] "Mutinomial Naive Bayes," [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes.

[10] "Support Vector Machine," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[11] "Fasttext," [Online]. Available: https://fasttext.cc/.

[12] "Word embeddings count word2vec," [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/.

## PHỤ LỤC 1: SOURCE CODE

Source code của đồ án được lưu trên github theo link dưới.

Link github: https://github.com/NguyenXuanVinh2000/NLP-Classifcation.git