ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ

PHÂN LOẠI CẢM XÚC DỰA TRÊN VĂN BẢN

Tiểu luận môn học

Học phần: Học máy

(Lớp: K66 - Kỹ thuật Điện tử và Tin học)

Hà Nội - 2024

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ

PHÂN LOẠI CẢM XÚC DỰA TRÊN VĂN BẢN

Tiểu luận môn học

Học phần: Học máy

(Lớp: K66 - Kỹ thuật Điện tử và Tin học)

Giảng viên: TS. Phạm Tiến Lâm

TS. Nguyễn Tiến Cường

Hà Nội - 2024

THÀNH VIÊN NHÓM

1. Lê Trung Kiên - 21002214
2. Lê Sơn Tùng - 21002243
3. Phạm Thu Thủy - 21002240
4. Nguyễn Mỹ Anh - 21002187

Lời cảm ơn

Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo. Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian. Trong chương trình học, chúng tôi đã được tiếp cận môn học này và đã được giảng dạy những kiến thức của môn học, ứng dụng của chúng trong đời sống. Rõ ràng học máy có vai trò quan trọng trong sự phát triển của con người trong đời sống ngày nay.

Trong quá trình hoàn thành bài báo cáo này, đầu tiên xin cho phép nhóm được gửi lời cảm ơn đến thầy TS. Phạm Tiến Lâm và thầy TS. Nguyễn Tiến Cường. Trong quá trình giảng dạy của môn học, các thầy không chỉ là người truyền đạt kiến thức về môn học mà còn là người truyền cảm hứng với mỗi thành viên của nhóm về môn học này. Bên cạnh đó, chúng tôi xin được cảm ơn thầy Nguyễn Việt Anh và thầy Nguyễn Viết Thắng với những tài liệu cũng như kiến thức trong các giờ thực hành trên lớp. Lời cuối cùng, xin được cảm ơn tất cả các thành viên trong nhóm đã không ngại khó khăn, luôn đoàn kết cùng hỗ trợ lẫn nhau để hoàn thành đề tài này.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh sách tên viết tắt

BoW	**B**ag **o**f **W**ord
TF - IDF	**T**erm **F**requency - **I**nvert **D**ocument **F**requency
SVM	**S**upport **V**ector **M**achine
RF	**R**andom **F**orest
CNN	**C**onvolution **N**eural **N**etwork
LSTM	**L**ong **S**hort - **T**erm **M**emory

MỞ ĐẦU

Thể hiện cảm xúc là nhu cầu cơ bản của con người, mỗi người sử dụng ngôn ngữ không chỉ để giao tiếp mà bày tỏ cảm xúc của mình. Paul Ekman đã đề xuất sáu cảm xúc cơ bản của con người bao gồm thích thú, buồn, giận dữ, ngạc nhiên, sợ hãi và ghê tởm qua nét mặt. Tuy nhiên, ngoài nét mặt, nhiều nguồn thông tin khác nhau có thể được sử dụng để phân tích cảm xúc vì vậy nhận dạng cảm xúc đã nổi lên như một lĩnh vực nghiên cứu quan trọng.

Trong những năm gần đây, nhận dạng cảm xúc trong văn bản đã trở nên phổ biến hơn do tiềm năng ứng dụng rộng lớn của nó trong tiếp thị, bảo mật, tâm lý học, tương tác giữa con người với máy tính, trí tuệ nhân tạo, v.v. Việc xây dựng hệ thống nhận diện cảm xúc với văn bản tiếng Việt là một bước tiến lớn trong xử lý ngôn ngữ tự nhiên, giúp giải quyết được nhiều vấn đề đang mắc phải.

Đề tài này tập trung vào vấn đề nhận biết cảm xúc đối với các bình luận tiếng Việt trên mạng xã hội. Cụ thể hơn, đầu vào của bài toán là một bình luận tiếng Việt từ mạng xã hội, và đầu ra là một cảm xúc được gắn nhãn trước đó, là một trong các cảm xúc: “enjoyment, sadness, anger, surprise, fear, disgust và other”. Trong đó, mỗi bình luận diễn đạt cảm xúc từ người dùng được biểu diễn thành một vector để đưa vào huấn luyện mô hình phân lớp. Hai mô hình học máy được triển khai trong đề tài bao gồm Support Vector Machine (SVM) và Random Forest so sánh hiệu quả với hai mô hình deep learning bao gồm Convolutional Neural Network (CNN) và Long Short-Term Memory (LSTM).

Để làm rõ hơn về đề tài này, bài báo cáo được chia làm các nội dung như sau:

Chương 1: Tổng quan.

Chương 2: Các cơ sở lý thuyết.

Chương 3: Thực nghiệm.

Chuong 4: Kết quả.

Chương 5: Kết luận.

Chương 1 TỔNG QUAN

1.1 Ngôn ngữ tự nhiên

Ngôn ngữ tự nhiên là ngôn ngữ được con người sử dụng để giao tiếp hàng ngày, chẳng hạn như tiếng Việt, tiếng Anh, tiếng Trung,... Theo thống kê, trên thế giới có khoảng 5600 ngôn ngữ, được phân bố rất không đồng đều và chỉ có một số ít các ngôn ngữ là có chữ viết. Khác với ngôn ngữ lập trình hoặc ngôn ngữ hình thức, ngôn ngữ tự nhiên có cấu trúc phức tạp, linh hoạt và thường chứa nhiều sự mơ hồ.

Các đặc điểm chính của ngôn ngữ tự nhiên:

- Tính Biến Đổi: Ngôn ngữ tự nhiên không cố định mà thay đổi liên tục theo thời gian, văn hóa, và bối cảnh. Từ mới được tạo ra, nghĩa của từ có thể thay đổi, và cách sử dụng ngôn ngữ có thể biến đổi dựa trên yếu tố xã hội và lịch sử.
- Tính Đa Nghĩa (Ambiguity): Ngôn ngữ tự nhiên thường có nhiều từ hoặc câu có thể mang nhiều nghĩa khác nhau tùy thuộc vào ngữ cảnh. Ví dụ, từ "bank" trong tiếng Anh có thể chỉ ngân hàng hoặc bờ sông.
- Tính Ngữ Cảnh (Context): Ý nghĩa của từ và câu trong ngôn ngữ tự nhiên phụ thuộc nhiều vào ngữ cảnh. Ngữ cảnh bao gồm thông tin về tình huống giao tiếp, người nói, người nghe, và các yếu tố môi trường.
- Tính Không Hoàn Hảo (Imperfectness): Ngôn ngữ tự nhiên không phải lúc nào cũng rõ ràng và có thể chứa nhiều lỗi ngữ pháp, lỗi chính tả, hoặc cấu trúc câu phức tạp và không hoàn chỉnh, đặc biệt trong ngôn ngữ nói.
- Tính Hệ Thống (Systematic): Dù ngôn ngữ tự nhiên có vẻ ngẫu nhiên, nó vẫn tuân theo một số quy tắc ngữ pháp và cú pháp cụ thể. Các quy tắc này giúp cấu trúc ngôn ngữ trở nên dễ hiểu và có thể dự đoán được.
- Tính Đa Phương Thức (Multimodality): Ngôn ngữ tự nhiên không chỉ bao gồm ngôn từ mà còn có thể bao gồm các yếu tố phi ngôn ngữ như cử chỉ, nét mặt, ngữ điệu, và âm lượng, tất cả đều góp phần vào việc truyền tải thông điệp.

- Tính Hài Hòa (Arbitrariness): Mối quan hệ giữa từ và ý nghĩa của từ trong ngôn ngữ tự nhiên thường là tùy ý. Không có lý do tự nhiên tại sao một đối tượng cụ thể lại được gọi bằng một từ cụ thể.
- Tính Phân Cấp (Hierarchical Structure): Ngôn ngữ tự nhiên có cấu trúc phân cấp, từ âm vị đến từ vựng, cụm từ, câu, và đoạn văn. Mỗi cấp độ này có các quy tắc riêng biệt.

Những đặc điểm này làm cho ngôn ngữ tự nhiên trở nên phức tạp và phong phú, tạo ra nhiều thách thức nhưng cũng đồng thời cung cấp nhiều cơ hội cho nghiên cứu và ứng dụng trong các lĩnh vực như ngôn ngữ học, xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), và trí tuệ nhân tạo. Việc hiểu rõ các đặc điểm này có thể giúp con người phát triển các công nghệ và mô hình tốt hơn để xử lý và tương tác với ngôn ngữ tự nhiên.

1.2 Ngôn ngữ tiếng Việt

Tiếng Việt là ngôn ngữ đơn lập, nghĩa là trong mỗi âm tiết đều được phát âm tách rời nhau và được biểu diễn bằng một chữ viết cụ thể. Đặc điểm này được thể hiện ở tất cả các mặt như về ngữ âm, từ vựng, ngữ pháp. Từ điển từ tiếng Việt (Vietlex) bao gồm trên 40.000 từ, trong đó:

- 81.55% âm tiết là từ: từ đơn.
- 15.69% các từ trong từ điển là từ đơn.
- 70.72% từ ghép có 2 âm tiết.
- 13.59% từ ghép trên 3 âm tiết.
- 1.04% từ ghép trên 4 âm tiết.

Việc sắp xếp các từ trong tiếng Việt theo một trật tự nhất định sẽ mang ý nghĩa khác nhau qua đó biểu thị các quan hệ cú pháp, ví dụ: “Mùa xuân lại đến” khác với “Lại đến mùa xuân“. Nhờ kết hợp trật tự của từ mà ngữ nghĩa của chúng cũng khác nhau. Trong

tiếng Việt, trật tự kết cấu câu chủ ngữ đứng trước, vị ngữ đứng sau là trật tự phổ biến nhất.

1.3 Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (NLP - Natural Language Processing) là một lĩnh vực trong trí tuệ nhân tạo và khoa học máy tính tập trung vào việc tương tác giữa máy tính và ngôn ngữ tự nhiên của con người. Mục tiêu của NLP là cho phép máy tính hiểu, diễn giải và tạo ra ngôn ngữ của con người một cách có ý nghĩa.

Các Kỹ Thuật và Công Cụ trong NLP:

- **Tiền xử lý (Preprocessing)**

  Tách từ (Tokenization): Chia văn bản thành các đơn vị nhỏ hơn như từ hoặc cụm từ.

  Xử lý stop word (Stop Words Removal): Loại bỏ các từ không mang nhiều ý nghĩa như "là", "và", "nhưng".

  Stemming and Lemmatization: Chuyển đổi các từ về dạng gốc của chúng.

- **Mô Hình Ngôn Ngữ (Language Models)**

  Bag of Words (BoW): Biểu diễn văn bản dưới dạng tập hợp các từ và tần suất xuất hiện của chúng.

  TF-IDF (Term Frequency-Inverse Document Frequency): Đánh giá tầm quan trọng của từ trong văn bản so với toàn bộ tập văn bản.

  Word Embeddings: Biểu diễn từ dưới dạng vector số học trong không gian liên tục, ví dụ như Word2Vec, GloVe, FastText.

- **Mô Hình Học Sâu (Deep Learning Models)**

  RNN (Recurrent Neural Networks): Mô hình mạng nơ-ron hồi quy, phù hợp với dữ liệu tuần tự như văn bản.

  LSTM (Long Short-Term Memory): Một loại RNN cải tiến, khắc phục vấn đề vanishing gradient.

1.4 Mô hình dự án

Trong đề tài này, trước tiên một file dữ liệu thô được sử dụng, chưa được xử lý bao gồm các câu bình luận của người dùng mạng xã hội tại Việt Nam. Mỗi câu đã được gán nhãn trước khi tiến hành học máy. Dữ liệu được chia thành ba nhóm: tập dữ liệu huấn luyện (training data), tập dữ liệu kiểm tra (test data), tập kiểm thử (validation data).

Giai đoạn huấn luyện: là giai đoạn học tập trên tập dữ liệu huấn luyện của mô hình phân loại cảm xúc trong văn bản. Ở bước này, mô hình sẽ học từ dữ liệu có nhãn (trong dự án này nhãn là enjoyment, sadness, anger, surprise, fear, disgust và other). Dữ liệu văn bản sẽ được số hóa thông qua bộ trích xuất đặc trưng để mỗi mẫu dữ liệu trong tập huấn luyện trở thành 1 vector nhiều chiều. Thuật toán máy học sẽ học và tối ưu các tham số để đạt được kết quả tốt trên tập dữ liệu này. Nhãn của dữ liệu được dùng để đánh giá việc mô hình học tốt hay không và dựa vào đó để tối ưu.

Giai đoạn kiểm tra: là giai đoạn sử dụng mô hình học máy sau khi nó đã học xong. Ở giai đoạn này, dữ liệu trên tập dữ liệu kiểm tra cần dự đoán cũng vẫn thực hiện các bước trích xuất đặc trưng. Mô hình đã học sau đó nhận đầu vào là đặc trưng đó và đưa ra kết quả dự đoán. Kết quả phân lớp đầu ra sẽ được ghi nhận lại để so sánh với nhãn mong đợi ban đầu của dữ liệu, từ đó cho chúng tôi thu được kết quả độ chính xác của mô hình

Chương 2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu

Bài toán nhận diện cảm xúc thuộc dạng bài toán phân tích ngữ nghĩa văn bản. Bài toán được giải quyết bằng cách phát triển một mô hình để phân tích và hiểu được ý nghĩa của câu văn, đoạn văn để quyết định xem câu văn đó hay đoạn văn đó mang ý nghĩa sắc thái cảm xúc nào. Về cơ bản, có thể chia cảm xúc con người thành nhiều loại và việc này tương ứng với các bài toán phân lớp dữ liệu trong khai thác dữ liệu. Đề tài này xây dựng ứng dụng nhận diện cảm xúc người dùng bằng phương pháp phân lớp dữ liệu.

Bằng cách mô tả khái quát mô hình phân tích cảm xúc từ bình luận của người dùng mạng xã hội: Dữ liệu đầu vào của bài toán là một câu văn, đoạn văn hay tổng quát hơn là một văn bản, còn kết quả đầu ra mong muốn là loại cảm xúc nào. Chẳng hạn với bài toán đánh giá bình luận, có thể phân loại cảm xúc người dùng ở 6 mức độ có tính chất định tính: enjoyment, sadness, anger, surprise, fear, disgust và other.

2.2 Phương pháp Word Embedding

Word embedding là một trong những bước quan trọng khi xây dựng bài toán phân tích cảm xúc trong văn bản tiếng Việt bằng mô hình máy học. Một lý do cơ bản cho việc vector hóa văn bản là máy tính không thể hiểu được nghĩa của các từ. Như vậy để xử lý ngôn ngữ tự nhiên 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 các văn bản thành các vector. Khi đó các từ hay các cụm từ được ánh xạ thành những vector trong không gian số thực. Hai phương pháp được sử dụng cho việc vector hóa văn bản bao gồm:

- Phương pháp Word Embedding cổ điển
- Phương pháp Neural Embedding (Vector hóa văn bản theo phương pháp mạng nơ-ron).

2.2.1 Word Embedding cổ điển

Bag of Words

Bag of Words (BoW) là một phương pháp để trích xuất các đặc điểm từ các dữ liệu văn bản. Các đặc điểm này có thể được sử dụng để đào tạo các thuật toán học máy. Nó tạo ra một kho từ vựng chứa tất cả các từ duy nhất có trong kho ngữ liệu của tập huấn luyện. Hay nói cách khác, đó là một tập hợp bao gồm các cặp giá trị key và value, giá trị key là từ duy nhất có trong tập dữ liệu, giá trị value là số lần xuất hiện của từ đó trong câu, và BoW hầu như không quan tâm đến thứ tự xuất hiện của các từ đó.

Ví dụ với hai câu sau:

- "Kiên thích chơi game, Thủy không thích chơi game."
- "Kiên cũng thích xem bóng đá."

Các câu này sẽ được biểu diễn ở dạng tập hợp các từ như sau:

- ["Kiên", "thích", "chơi", "game", "Thủy", "không", "thích", "chơi", "game."]
- ["Kiên", "cũng", "thích", "xem", "bóng", "đá."]

Ngoài ra, trong mỗi câu, ta sẽ thực hiện loại bỏ số lần xuất hiện lặp lại của từ và sử dụng số lượng từ để biểu diễn.

- {"Kiên": 1, "thích": 2, "chơi": 2, "game": 2, "Thủy":1, "không":1}
- {"Kiên": 1, "cũng": 1, "thích": 1, "xem": 1, "bóng": 1, "đá": 1}

Giả sử những câu này là một phần của dữ liệu văn bản, dưới đây là tần suất từ được kết hợp cho toàn bộ dữ liệu văn bản.

- {"Kiên": 2, "thích": 3, "chơi": 2, "game": 2, "Thủy": 1, "không": 1, "cũng": 1, "xem": 1, "bóng": 1, "đá": 1}

Cấu trúc từ vựng ở trên của tất cả các từ, với số lượng từ tương ứng của chúng, sẽ được sử dụng để tạo các vectơ cho mỗi câu. Độ dài của vectơ sẽ luôn bằng kích thước của tập hợp từ vựng. Trong trường hợp này, độ dài vectơ là 10. Để biểu diễn các câu gốc trong một vectơ, mỗi vectơ sẽ được khởi tạo bằng tất cả các giá trị số không - [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. Tiếp theo là lặp lại và so sánh với từng từ trong tập hợp từ vựng của chúng ta và tăng giá trị vectơ nếu câu có chứa từ đó.

"Kiên thích chơi game. Thủy không thích chơi game." [1, 2, 2, 2, 1, 1, 0, 0, 0, 0]

"Kiên cũng thích xem bóng đá." [1, 0, 0, 0, 0, 0, 1, 1, 1, 1]

Ví dụ, trong câu 1, từ "thích" xuất hiện ở vị trí thứ hai và xuất hiện hai lần. Vì vậy, phần tử thứ hai của vectơ cho câu 1 sẽ là 2: [1, 2, 2, 2, 1, 1, 0, 0, 0, 0]. Trong phương pháp BoW này, các từ giống nhau sẽ có trọng số như nhau. Nó không quan tâm đến tần suất xuất hiện của từ hay ngữ cảnh của từ. Trong thực tế, khi phân tích từ cần hiểu rõ nghĩa của từ, xét nghĩa của từ trong toàn văn bản hơn là xét nghĩ độc lập.

TF – IDF

TF-IDF là một kĩ thuật rất nổi tiếng, được sử dụng trong nhiều bài toàn NLP và khai phá dữ liệu dạng văn bản với mục đích: tính weight (độ quan trọng) của từ trong một văn bản cụ thể, văn bản đó nằm trong một tập nhiều văn bản khác nhau (corpus).

TF-IDF thể hiện được trọng số của mỗi từ theo ngữ cảnh trong toàn văn bản. Nó chuyển đổi dạng biểu diễn văn bản thành dạng không gian vector (VSM), hoặc thành những vector thưa thớt. Phương pháp này giúp làm tăng tỷ lệ thuận với số lần xuất hiện của từ trong văn bản và số các văn bản mà có chứa các từ đó trên toàn tập liệu đầu vào.

TF(Term frequency) : Tần suất xuất hiện của một từ trong một đoạn văn bản. IDF(Invert Document Frequency) : Được dùng để đánh giá mức độ quan trọng của một từ trong văn bản.

Khi tính TF thì mức độ quan trọng của các từ là như nhau. Tuy nhiên trong văn bản thường xuất hiện nhiều từ không quan trọng với tần xuất cao (ví dụ stop word). Do đó cần làm giảm đi mức độ quan trọng của từ đó bằng IDF. Cách tính TF-IDF được cho bởi công thức sau:

$$TF_i = \frac{n_i}{N_i} \tag{2.1}$$

Trong đó:

- i: 1...D
- n_i: Tần số xuất hiện của từ trong văn bản i.
- N_i : Tổng số từ trong văn bản i.

$$IDF_i = log(\frac{D}{d}) \tag{2.2}$$

Trong đó:

- D : Tổng số document trong tập dữ liệu.
- d : Số lương document có sự xuất hiện của từ.

Khi đó:

$$TF_i - IDF_i = TF_i \times IDF_i \tag{2.3}$$

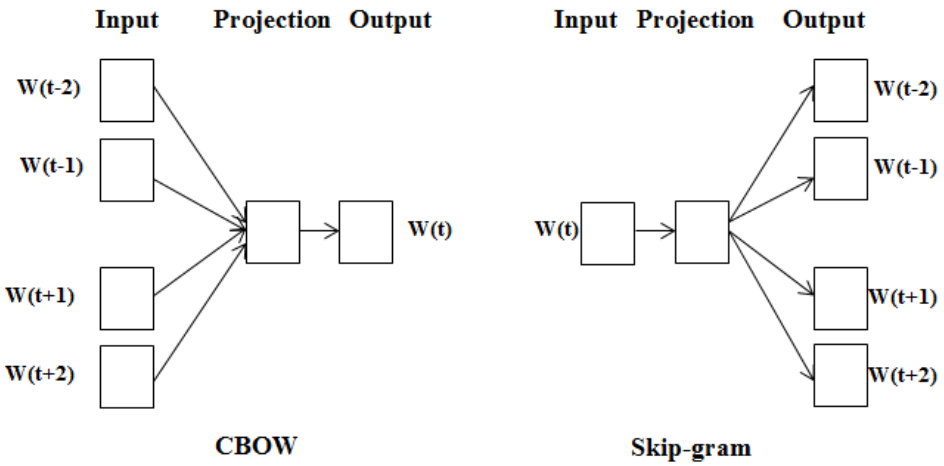
2.2.2 Neural Embedding

Word2Vec

Phương pháp Word2vec là một nhóm các mô hình tạo ra các biểu diễn embedding của từ trong một không gian có số chiều thấp hơn nhiều lần so với số từ trong từ điển bằng cách sử dụng mạng nơ-ron nông, hai lớp. Mô hình dự đoán sẽ biểu diễn vector từ thông qua các từ, ngữ cảnh xung quanh nhằm tăng khả năng dự đoán được nghĩa của từ.

Có hai cách xây dựng mô hình Word2vec dùng để biểu diễn phân tán của từ trong không gian vector:

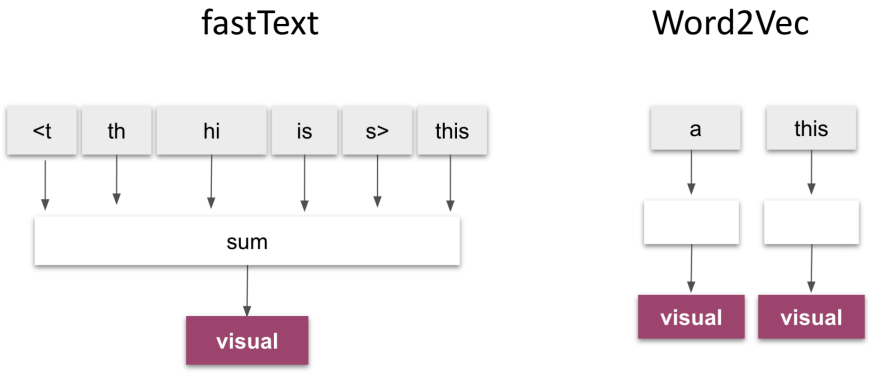
- Continuous bag-of-words (CBOW): Sử dụng từ ngữ cảnh để dự đoán từ mục tiêu. Thứ tự của các từ ngữ cảnh không ảnh hưởng đến dự đoán (giả định bag-of-words).
- Continuous skip-gram: Sử dụng từ hiện tại để dự đoán các từ ngữ cảnh. Continuous skip-gram chú trọng các từ ngữ cảnh ở phạm vi gần. Mỗi vectơ ngữ cảnh được cân nhắc và so sánh độc lập với CBOW.



Hình 2.1: Mô hình CBOW

Trong hai thuật toán trên, thuật toán CBOW khi thực thi sẽ ít tốn thời gian để huấn luyện mô hình hơn Skip-gram. Tuy nhiên, thuật toán Skip-gram có độ chính xác cao hơn và có chứa cả những từ ít xuất hiện.

FastText



Hình 2.2: Mô hình fastText và Word2Vec

FastText được xây dựng trên Word2Vec bằng cách học các biểu diễn vectơ cho mỗi từ và n-gam được tìm thấy trong mỗi từ. Giá trị của các biểu diễn sau đó được tính trung bình thành một vectơ ở mỗi bước huấn luyện. Mặc dù bổ sung nhiều tính toán cho việc đào tạo, nó cho phép nhúng từ để mã hóa thông tin từ phụ. Các vectơ FastText đã được chứng minh là chính xác hơn các vectơ Word2Vec bằng một số biện pháp khác nhau.

2.3 Các mô hình học máy

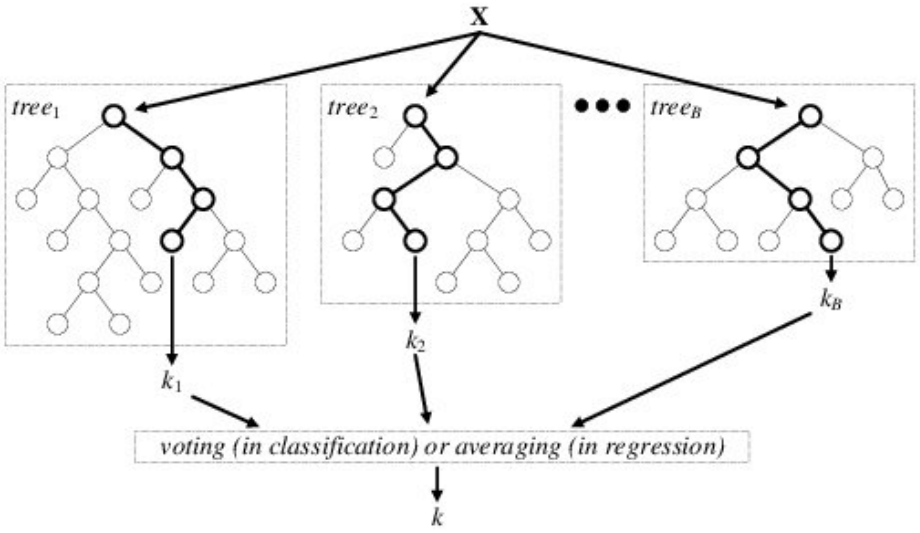
2.3.1 Mô hình Random Forest

Decision Tree

Khi xây dựng cây quyết định, thuật toán sẽ lặp qua tất cả các đặc trưng và tính toán Information Gain cho mỗi đặc trưng, sau đó chọn các đặc trưng có Information Gain cao nhất để phân chia dữ liệu tại node hiện tại. Information Gain đo lường sự giảm entropy (độ không chính xác) của dữ liệu sau khi được phân chia bởi một đặc trưng cụ thể. Cụ thể, nó đo lường sự "thuần túy" (purity) của các nhãn trong dữ liệu. Entropy càng thấp tức là dữ liệu càng được phân loại một cách chính xác và rõ ràng.

Random forest

Random Forest (RF) là một thuật toán học máy linh hoạt khi được sử dụng cho các bài toán phân loại, dự đoán giá trị hồi quy tuyến tính và các tác vụ đa đầu ra. Ý tưởng của RF là sử dụng một tập hợp Decision Tree (Cây quyết định), mỗi phân loại được huấn luyện trên các phần khác nhau của tập dữ liệu. Sau đó RF sẽ lấy kết quả phân loại dựa trên các cây phân loại.



Hình 2.3: Mô hình Random Forest

Thay vì sử dụng tất cả các đặc trưng để xây dựng cây quyết định, Random Forest sẽ chọn một số lượng ngẫu nhiên các đặc trưng để xem xét tại mỗi node.

Mô hình Random Forest:

- Tạo mẫu ngẫu nhiên (Boostrapping):

  Từ bộ dữ liệu huấn luyện gốc, tạo ra nhiều tập con dữ liệu bằng cách lấy mẫu ngẫu nhiên với phép lặp (có thể lấy lại một phần tử đã chọn).

  Khi tập mẫu được rút ra từ một tập huấn luyện của một cây với sự thay thế (bagging), thì theo ước tính có khoảng 1/3 các phần từ không có nằm trong mẫu này. Điều này có nghĩa là chỉ có khoảng 2/3 các phần tử trong tập huấn luyện tham gia vào trong các tính toán của chúng ta, và 1/3 các phần tử này được gọi là dữ liệu out-of-bag. Dữ liệu out-of-bag được sử dụng để ước lượng lỗi tạo ra từ việc kết hợp các kết quả từ các cây tổng hợp trong Random Forest cũng như dùng để ước tính độ quan trọng thuộc tính (variable important).

- Xây dựng cây quyết định (Decision Tree)

  Chọn Ngẫu Nhiên Các Đặc Trưng (Feature Selection): Tại mỗi nút của cây, chọn ngẫu nhiên một tập hợp con nhỏ các đặc trưng để tìm ra đặc trưng tốt nhất để chia nhánh. Điều này giúp giảm tương quan giữa các cây quyết định.

Chia Dữ Liệu: Sử dụng tiêu chí như Gini impurity hoặc thông tin entropy để tính toán Information Gain và tìm ra cách chia tốt nhất dựa trên các đặc trưng đã chọn.

- Lặp lại quá trình tạo mẫu ngẫu nhiên và xây dựng cây quyết định cho đến khi đạt được số lượng cây quyết định mong muốn trong rừng.

- Voting hoặc Averaging: Khi thực hiện dự đoán, Random Forest sử dụng phương pháp voting (trong trường hợp phân loại) hoặc averaging (trong trường hợp hồi quy) từ các cây thành viên. Điều này giúp tạo ra một mô hình tổng hợp mạnh mẽ và giảm variance.

- Đánh gia mô hình:

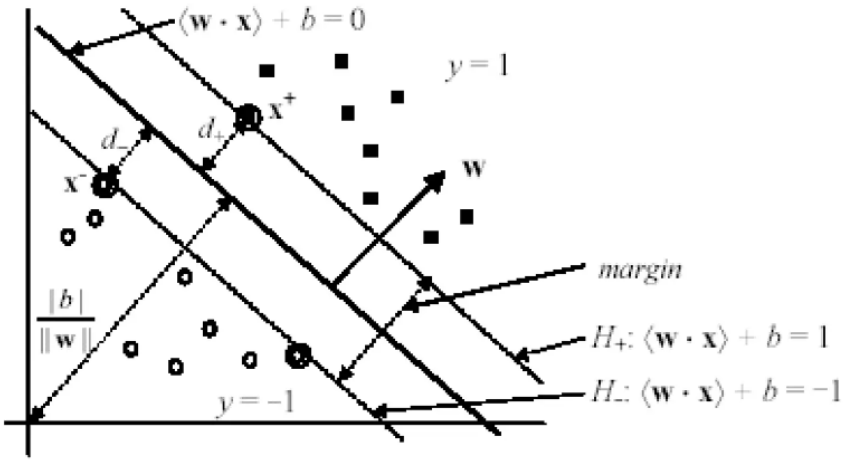
  Đánh Giá Hiệu Suất: Sử dụng các phương pháp đánh giá như accuracy, precision, recall, F1-score cho bài toán phân loại; và RMSE, MAE cho bài toán hồi quy.

  K-fold Cross Validation: Thực hiện k-fold cross validation để đánh giá mô hình một cách toàn diện hơn.

2.3.2 Mô hình SVM

SVM (Support Vector Machine) là một thuật toán học máy có giám sát, được sử dụng chủ yếu cho các bài toán phân loại và hồi quy. Đặc điểm chính của SVM là khả năng tìm ra một siêu phẳng tối ưu để phân tách dữ liệu thuộc các lớp khác nhau trong không gian nhiều chiều.

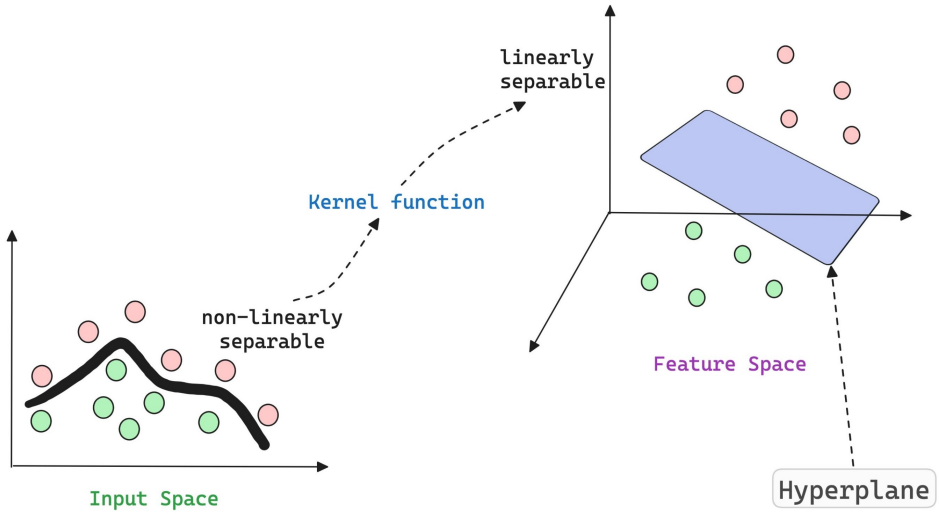
Trong không gian nhiều chiều, một siêu phẳng là một không gian con có chiều thấp hơn. Ví dụ, trong không gian hai chiều, siêu phẳng là một đường thẳng; trong không gian ba chiều, siêu phẳng là một mặt phẳng.



Hình 2.4: Mô hình SVM tuyến tính

Với SVM, mục tiêu tối ưu hóa là tìm ra một siêu phẳng (hyperplane) phân chia các điểm dữ liệu thành các lớp khác nhau sao cho khoảng cách từ các điểm dữ liệu gần nhất đến siêu phẳng là lớn nhất có thể. Cụ thể, cần tối thiểu hóa độ lớn của vectơ trọng số w, được biểu diễn bằng $||w||^2$, để đảm bảo siêu phẳng được xác định bởi w càng "phẳng" càng tốt.

Để tối thiểu hóa module w trong SVM, cần tìm cách tối ưu hóa hàm mất mát (loss function) của SVM, thường được gọi là hàm mất mát hinge (hinge loss function). Mục tiêu là tìm ra một vectơ trọng số w và một độ lệch b (bias) sao cho hàm mất mát được tối thiểu hóa và mô hình SVM hoạt động hiệu quả trong việc phân loại dữ liệu. Một phương pháp phổ biến để tối thiểu hóa w trong SVM là sử dụng các thuật toán tối ưu hóa như Gradient Descent hoặc các biến thể của nó: SGD, SMO, QP, . . .



Hình 2.5: Mô hình SVM phi tuyến tính

Khi dữ liệu không thể phân tách bằng một siêu phẳng trong không gian gốc, SVM sử dụng các hàm nhân (Kernel function) để ánh xạ dữ liệu sang không gian chiều cao hơn, nơi có thể phân tách được bằng siêu phẳng. Các hàm nhân phổ biến bao gồm:

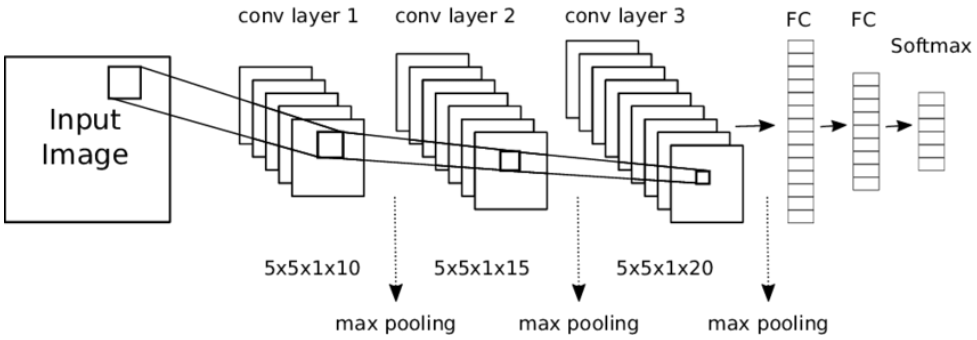
- Hàm nhân tuyến tính (Linear Kernel).
- Hàm nhân Gaussian hoặc RBF (Radial Basis Function).
- Hàm nhân đa thức (Polynomial Kernel).

2.4 Các mô hình mạng neuron dùng trong học sâu

2.4.1 Mô hình CNN

Mạng thần kinh tích chập (CNN - Convolution Neural Network) là một loại mô hình học máy, cụ thể là một loại thuật toán học sâu rất phù hợp để phân tích dữ liệu trực quan. CNN - đôi khi được gọi là mạng convnet - sử dụng các nguyên tắc từ đại số tuyến tính, đặc biệt là các phép toán tích chập, để trích xuất các đặc điểm và xác định các mẫu trong hình ảnh. Mặc dù CNN chủ yếu được sử dụng để xử lý hình ảnh nhưng chúng cũng có thể được điều chỉnh để hoạt động với âm thanh và dữ liệu tín hiệu khác.

Cấu trúc của CNN



Hình 2.6: Mô hình CNN

Một CNN thường bao gồm một số lớp, có thể được phân loại thành ba nhóm: lớp tích chập, lớp gộp và lớp được kết nối đầy đủ. Khi dữ liệu đi qua các lớp này, độ phức tạp của CNN tăng lên, cho phép CNN liên tục xác định các phần lớn hơn của hình ảnh và nhiều tính năng trừu tượng hơn.

Convolution layer:

- Lớp tích chập là khối xây dựng cơ bản của CNN và là nơi diễn ra phần lớn các phép tính. Lớp này sử dụng bộ lọc hoặc hạt nhân - một ma trận trọng số nhỏ - để di chuyển qua trường tiếp nhận của hình ảnh đầu vào nhằm phát hiện sự hiện diện của các tính năng cụ thể.
- Quá trình bắt đầu bằng cách trượt hạt nhân theo chiều rộng và chiều cao của hình ảnh, cuối cùng quét toàn bộ hình ảnh qua nhiều lần lặp. Tại mỗi vị trí, một tích số chấm được tính giữa trọng số của hạt nhân và giá trị pixel của hình ảnh bên dưới hạt nhân. Điều này biến đổi hình ảnh đầu vào thành một tập hợp các bản đồ đặc trưng hoặc các đặc điểm phức tạp, mỗi đặc điểm thể hiện sự hiện diện và cường độ của một đặc điểm nhất định tại các điểm khác nhau trong hình ảnh.
- CNN thường bao gồm nhiều lớp chập xếp chồng lên nhau. Thông qua kiến trúc phân lớp này, CNN diễn giải dần dần thông tin hình ảnh có trong dữ liệu hình ảnh thô. Trong các lớp trước, CNN xác định các đặc điểm cơ bản như cạnh, kết cấu hoặc màu sắc. Các lớp sâu hơn nhận đầu vào từ bản đồ đặc trưng của các lớp trước, cho phép chúng phát hiện các mẫu, vật thể và cảnh phức tạp hơn.

Pooling layer:

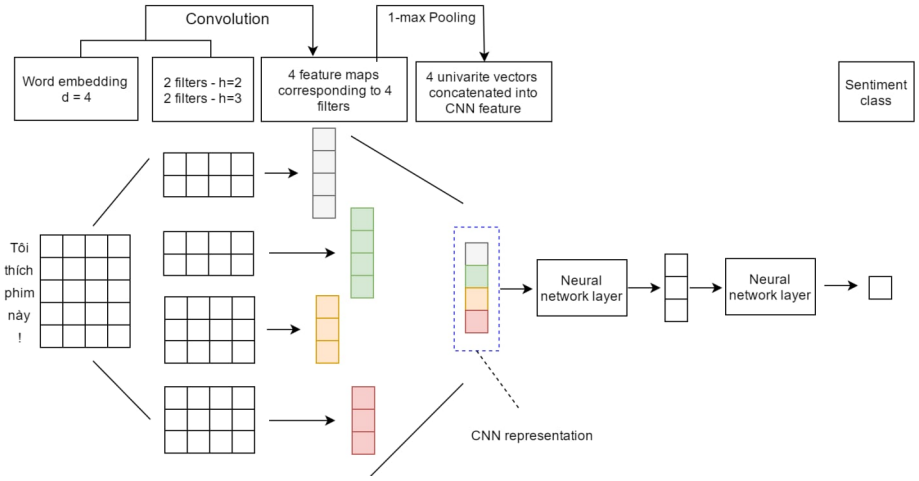
- Lớp gộp của CNN là một thành phần quan trọng theo sau lớp tích chập. Tương tự như lớp tích chập, các hoạt động của lớp gộp liên quan đến quá trình quét trên hình ảnh đầu vào, nhưng chức năng của nó thì khác.
- Lớp tổng hợp nhằm mục đích giảm tính chiều của dữ liệu đầu vào trong khi vẫn giữ lại thông tin quan trọng, do đó cải thiện hiệu quả tổng thể của mạng. Điều này thường đạt được thông qua việc lấy mẫu xuống: giảm số lượng điểm dữ liệu trong đầu vào.
- Đối với CNN, điều này thường có nghĩa là giảm số lượng pixel được sử dụng để thể hiện hình ảnh. Hình thức gộp phổ biến nhất là gộp tối đa, giữ lại giá trị tối đa trong một cửa sổ nhất định (tức là kích thước hạt nhân) trong khi loại bỏ các giá trị khác. Một kỹ thuật phổ biến khác, được gọi là gộp trung bình, có cách tiếp cận tương tự nhưng sử dụng giá trị trung bình thay vì giá trị tối đa.
- iệc lấy mẫu xuống làm giảm đáng kể tổng số tham số và tính toán. Ngoài việc nâng cao hiệu quả, điều này còn tăng cường khả năng khái quát hóa của mô hình. Các mô hình ít phức tạp hơn với các tính năng cấp cao hơn thường ít bị trang bị quá mức – một hiện tượng xảy ra khi một mô hình nhận biết nhiễu và các chi tiết quá cụ thể trong dữ liệu huấn luyện của nó, ảnh hưởng tiêu cực đến khả năng khái quát hóa thông tin mới, chưa được nhìn thấy.
- Việc giảm kích thước không gian của việc biểu diễn có một nhược điểm tiềm ẩn, đó là mất một số thông tin. Tuy nhiên, chỉ học những đặc điểm nổi bật nhất của dữ liệu đầu vào thường là đủ cho các tác vụ như phát hiện đối tượng và phân loại hình ảnh.

Fully connected layer

- Lớp được kết nối đầy đủ đóng một vai trò quan trọng trong các giai đoạn cuối cùng của CNN, nơi nó chịu trách nhiệm phân loại hình ảnh dựa trên các đặc điểm được trích xuất ở các lớp trước đó. Thuật ngữ kết nối đầy đủ có nghĩa là mỗi nơ-ron trong

một lớp được kết nối với mỗi nơ-ron ở lớp tiếp theo, cho phép CNN xem xét đồng thời tất cả các tính năng khi đưa ra quyết định phân loại cuối cùng.

- Không phải tất cả các lớp trong CNN đều được kết nối đầy đủ. Bởi vì các lớp được kết nối đầy đủ có nhiều tham số, nên việc áp dụng phương pháp này trên toàn bộ mạng sẽ tạo ra mật độ không cần thiết, làm tăng nguy cơ trang bị quá mức và khiến mạng rất tốn kém khi đào tạo về bộ nhớ và tính toán. Việc giới hạn số lượng các lớp được kết nối đầy đủ sẽ cân bằng hiệu quả tính toán và khả năng khái quát hóa với khả năng tìm hiểu các mẫu phức tạp.



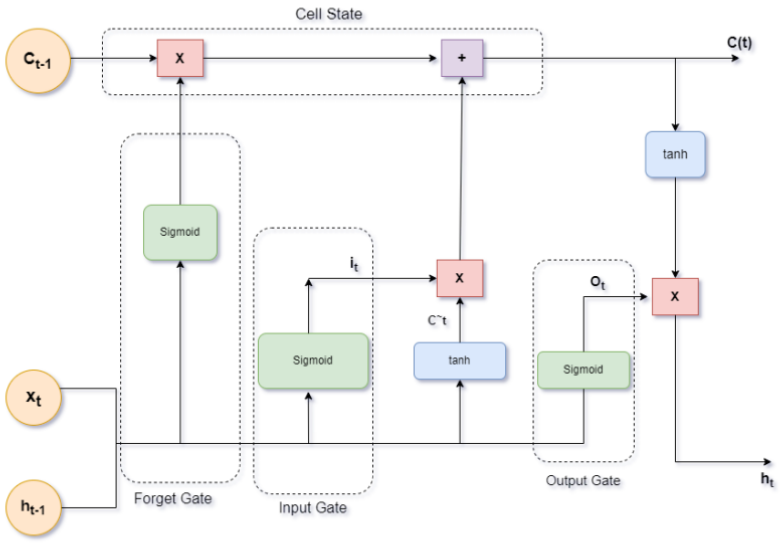
Hình 2.7: Mô hình CNN trong xử lý ngôn ngữ tự nhiên

2.4.2 Mô hình LSTM

Bộ nhớ ngắn hạn (LSTM - Long Short-Term Memory) LSTM cũng được áp dụng cho kho dữ liệu UIT-VSMEC vì nhiều lý do. Để bắt đầu, LSTM được coi là phương pháp tiên tiến nhất cho hầu hết các vấn đề dự đoán trình tự. Hơn nữa, qua hai cuộc thi WASSA-2018 và SemEval-2018 cho nhiệm vụ nhận dạng cảm xúc, chúng tôi ghi nhận LSTM được sử dụng hiệu quả nhất. Hơn nữa, LSTM có lợi thế hơn các mạng thần kinh thông thường và Mạng thần kinh tái phát (RNN) theo nhiều cách khác nhau nhờ đặc tính bộ nhớ chọn lọc trong thời gian dài. Do đó, chúng tôi quyết định sử dụng LSTM cho cùng một vấn đề trên kho dữ liệu của mình.

Kiến trúc LSTM: gồm một trạng thái ô (phần bộ nhớ của LSTM), cổng đầu vào, cổng đầu ra và cổng quên. Mỗi thành phần này có một vai trò cụ thể trong hoạt động của LSTM.

- Trạng thái ô (Cell State) : lưu trữ trạng thái của một chuỗi, do đó nó có khả năng giữ hoặc quên một số thông tin nhất định.
- Cổng đầu vào (Input Gate): quyết định thông tin mới nào được lưu trữ trong ô.
- Cổng đầu ra (Output Gate): xác định trạng thái ẩn tiếp theo.
- Cổng quên (Forget Gate): quyết định thông tin nào nên bỏ đi hoặc giữ lại.



Hình 2.8: Mô hình LSTM

Hoạt động của LSTM:

Giả sử chúng ta có một chuỗi các từ (w1, w2, w3, ..., wn) và chúng ta đang xử lý từng từ một trong chuỗi đó. Biểu thị trạng thái của LSTM tại bước thời gian t là (h_t, C_t), trong đó h_t là trạng thái ẩn và C_t là trạng thái ô.

Bước 1: LSTM nhận vectơ đầu vào (xt) và trạng thái trước đó (ht-1, Ct-1).

Bước 2: Cổng quên (f_t) quyết định thông tin nào cần loại bỏ khỏi trạng thái ô. Nó sử dụng vectơ đầu vào và trạng thái ẩn trước đó để tạo một số từ 0 đến 1 cho mỗi số ở trạng thái ô C_{t-1}. Giá trị gần 0 nghĩa là thông tin đó sẽ bị loại bỏ, còn giá trị gần 1 nghĩa là thông tin đó sẽ được giữ lại.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2.4}$$

Bước 3: Cổng đầu vào (i_t) quyết định thông tin mới nào sẽ được lưu trữ ở trạng thái ô. Nó có hai phần:

Lớp sigmoid được gọi là "lớp cổng đầu vào" quyết định giá trị nào sẽ cập nhật

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.5}$$

Lớp tanh tạo ra một vectơ gồm các trạng thái ô mới ($\tilde{C}_t$) có thể được thêm vào C_t

$$\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{2.6}$$

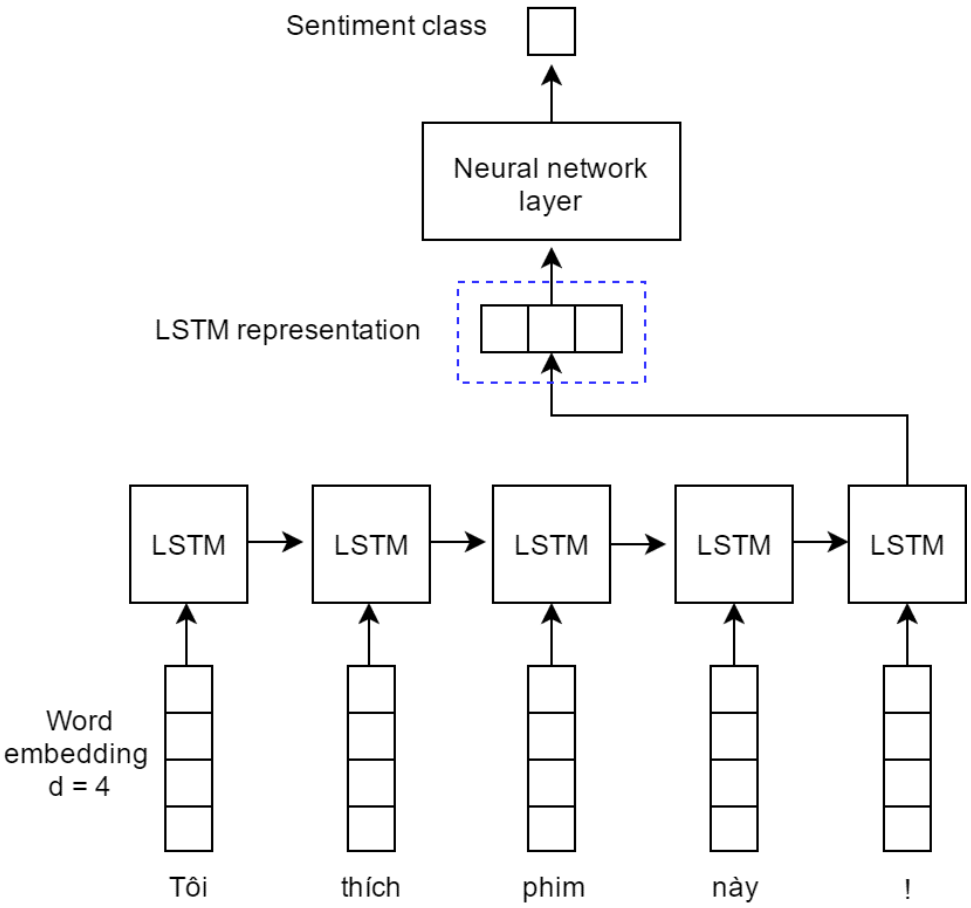
Bước 4: Cập nhật trạng thái ô cũ (C_{t-1}) sang trạng thái ô mới (C_t). Trạng thái ô cũ được nhân với ft để quên đi những điều chúng ta đã quyết định quên trước đó. Sau đó, chúng ta thêm các trạng thái ô mới, được điều chỉnh bởi i_t để tạo ra C_t

$$C_t = f_t.C_{t-1} + i_t * \tilde{C}_t \tag{2.7}$$

Bước 5: Quyết định đầu ra. Đầu tiên, chúng ta chạy một lớp sigmoid để quyết định phần nào của trạng thái ô mà chúng ta sẽ xuất ra. Sau đó, chúng ta đặt trạng thái ô thông

qua hàm tanh (để đặt các giá trị nằm trong khoảng từ -1 đến 1) và nhân nó với đầu ra o_t , chỉ xuất những phần mà chúng ta quyết định giữ lại.

$$h_t = o_t.tanh(C_t) \tag{2.8}$$



Hình 2.9: Mô hình LSTM trong xử lý ngôn ngữ tự nhiên

LSTM bao gồm bốn phần chính: Đầu vào nhúng từ, mạng di động LSTM, fully connected layer và softmax. Với đầu vào, mỗi ô trong mạng LSTM nhận được một vectơ từ được biểu thị bằng các từ nhúng có dạng [1 x n] trong đó n là độ dài cố định của câu. Sau đó, các ô tính toán các giá trị và nhận kết quả dưới dạng vectơ trong mạng di động LSTM. Các vectơ này sẽ được kết nối đầy đủ và các giá trị đầu ra sau đó sẽ chuyển qua hàm softmax để đưa ra phân loại phù hợp cho từng nhãn.

Chương 3 Thực nghiệm

3.1 Chuẩn bị dữ liệu

Để thực hiện được mô hình này đòi hỏi phải có được một tập dữ liệu càng lớn càng tốt và được gán nhãn đủ lớn để tao tập huấn luyện và tập kiểm tra bằng mô hình máy học có giám sát. Từ đó có thể đánh giá được độ chính xác thông qua mô hình.

	Unnamed: 0	Emotion	Sentence
0	188	Other	cho mình xin bài nhạc tên là gì với ạ
1	166	Disgust	cho đáng đời con quỷ . về nhà lôi con nhà mày ...
2	1345	Disgust	lo học đi . yêu đương lol gì hay lại thích học...
3	316	Enjoyment	ước gì sau này về già vẫn có thể như cụ này :))
4	1225	Enjoyment	mỗi lần có video của con là cứ coi đi coi lại ...
...	...	...	...

Hình 3.1: Mô tả bộ dữ liệu

Kho dữ liệu UIT-VSMEC là kho dữ liệu đầu tiên về nhận dạng cảm xúc cho văn bản trên mạng xã hội Việt Nam, được công khai bởi Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh . Với kho ngữ liệu UIT-VSMEC, chúng tôi có được 6.927 câu do con người chú thích với một trong bảy nhãn cảm xúc. Thống kê nhãn cảm xúc của kho ngữ liệu được trình bày ở bảng sau.

Nhãn cảm xúc	Số câu	Tỉ lệ (%)
Enjoyment	1965	28.36
Disgust	1338	19.31
Sadness	1149	16.59
Anger	480	6.72
Fear	395	5.7
Surprise	309	4.46
Other	1291	18.66
Tổng	6927	100

Bảng 3.1: Thống kê nhãn cảm xúc của kho dữ liệu UIT-VSMEC

Kho dữ liệu UIT-VSMEC được chia thành tỷ lệ 80:10:10, trong đó 80% kho ngữ liệu là tập train, 10% là tập validation, và phần còn lại là tập test. Kho dữ liệu UIT-VSMEC là một kho dữ liệu có nhãn không cân bằng, do đó, để đảm bảo rằng các câu trong nhãn có dung lượng thấp được phân bổ đầy đủ trong mỗi bộ, mô hình sử dụng phương pháp lấy mẫu phân tầng bằng cách sử dụng hàm train_test_split() được hỗ trợ bởi thư viện scikit learn để phân phối chúng thành các tập huấn luyện, xác thực và kiểm tra. Kết quả được trình bày trong bảng sau:

Nhãn cảm xúc	Tập train	Tập validation	Tập test	Tổng
Enjoyment	1573	205	187	1965
Disgust	1064	141	133	1338
Sadness	938	92	119	1149
Anger	395	38	47	480
Fear	317	38	47	395
Surprise	242	36	31	309
Other	1019	132	140	1291
Tổng	5548	686	693	6297

Bảng 3.2: Thống kê các câu gắn nhãn cảm xúc trong tập train, tập validation và tập test

3.2 Tiền xử lý

Tiền xử lý dữ liệu là một trong những bước quan trọng nhất khi giải quyết bất kỳ bài toán nào trong lĩnh vực Học máy. Để mô hình có thể đưa ra kết quả có độ chính xác cao thì bộ dữ liệu luôn cần được xử lý, làm sạch và biến đổi trước khi trở thành dữ liệu huấn luyện cho mô hình học máy.

3.2.1 Làm sạch văn bản (Text Cleaning)

Đối với project này, dữ liệu input đầu vào là các bình luận trên mạng xã hội Việt Nam. Dữ liệu thường không chuẩn, vì thế ta phải tiến hành làm sạch dữ liệu:

- Chuyển tất cả về chữ thường. Chữ in hoa và chữ thường sẽ có mã unicode khác nhau, về mặt ngữ nghĩa thì giống nhau nhưng máy tính sẽ không thể phân biệt dữ liệu đầu vào, dẫn đến kết quả dự đoán có thể bị ảnh hưởng. Vì vậy, việc chuyển

toàn bộ chữ về chữ thường là công việc hợp lý và cần thiết cho hệ thống phân tích và dự đoán

- Xóa bỏ những ký tự đặc biệt: con số, dấu câu, khoảng trắng, emoji vì đây là những thứ không có ý nghĩa gì trong câu.
- Xóa bỏ những kí tự lặp lại. VD: aaabbb -> ab.
- Xóa mọi dấu cách ở đầu hoặc cuối cũng như dấu chấm câu ở đầu hoặc cuối khỏi văn bản.
- Xóa dòng dữ liệu: tập dữ liệu thu về sẽ có rất nhiều dữ liệu bị trống, dữ liệu trống sẽ không có ý nghĩa trong quá trình phân tích, gây hao tốn bộ nhớ lưu trữ.
- Loại bỏ dấu của từ.
- Xử lý các từ đồng nghĩa và xử lý các từ sai chính tả: Hai công việc này cũng cần thiết trong quá trình tiền xử lý nhưng chúng tôi chưa nghĩ được thuật toán nào để sửa triệt để, chỉ có thể sửa thủ công.

3.2.2 Tách từ - Tokenization

Tiến hành xử lý bộ dữ liệu đã được làm sạch. Mô hình này sẽ áp dụng thuật toán Tokenziner - một nhánh con trong tập xử lý ngôn ngữ tự nhiên. Tokenziner cho phép ta vectơ hóa một kho ngữ liệu văn bản, bằng cách biến mỗi văn bản thành một chuỗi các số nguyên hoặc thành một vectơ trong đó hệ số cho mỗi mã thông báo có thể là nhị phân, dựa trên số từ, dựa trên tf-idf ...

Tách từ là một kỹ thuật cần thiết trong xử lý ngôn ngữ tự nhiên, nhiệm vụ chính là tách một đoạn text thành những “word” hoặc “token”. Tuy nhiên tiếng việt có nhiều từ ghép (Các từ phải đi cùng với nhau mới có ý nghĩa). Cho nên việc tách từ làm sao để không tách ra từ ghép là một việc cần thiết để tránh mất đi ý nghĩa của từ trong câu. Ở Việt Nam có một số thư viện có thể giúp thực hiện công việc này như pyvi, vntokenizer hay underthesea.

Sau việc tiền xử lý dữ liệu gồm tách từ và loại bỏ các hư từ cũng như các dấu câu không cần thiết. Dữ liệu thu được như sau:

[37]:

	Unnamed: 0	Emotion	Sentence	corpus
0	188	Other	cho mình xin bài nhạc tên là gì với ạ	cho mình xin bài nhạc tên là gì với ạ
1	166	Disgust	cho đáng đời con quỷ . về nhà lôi con nhà mày ...	cho đáng đời con quỷ về nhà lôi con nhà mày ra...
2	1345	Disgust	lo học đi . yêu đương lol gì hay lại thích học...	lo học đi yêu đương_lol gì hay lại thích học_s...
3	316	Enjoyment	ước gì sau này về già vẫn có thể như cụ này :))	ước gì sau_này về già vẫn có_thể như cụ này
4	1225	Enjoyment	mỗi lần có video của con là cứ coi đi coi lại ...	mỗi lần có video của con là cứ coi đi coi lại ...
5	1220	Anger	thằng kia sao mày bắt vợ với bỏ tao dọn thế ki...	thằng kia sao mày bắt vợ với bỏ_tao dọn thế ki...
6	44	Other	một lí do trog muôn vàn lí do	một lí_do trog_muôn vàn_lí_do
7	1951	Surprise	thật hay đùa ác vậy . không thể tin được	thật hay đùa ác vậy không_thể tin được
8	1249	Anger	ko phải con mình , mà xem còn thấy đau như vậy...	ko phải con mình mà xem còn thấy đau như_vậy h...
9	1063	Sadness	per nghe đi rồi khóc 1 trận cho thoải mái . đừ...	per nghe đi rồi khóc 1 trận cho thoải_mái đừng...
10	1523	Enjoyment	công nhận sáng tạo thật đấy	công_nhận sáng_tạo thật đấy
11	1785	Disgust	đòn tấn công cực gắt và cục súc của anh 😂😂😂	đòn tấn_công cực gắt và cục_súc của anh
12	1069	Other	trời nắng nóng thế này mình muốn bán nước khôn...	trời nắng_nóng thế_này mình muốn bán nước khôn...
13	776	Enjoyment	mình biết nữa ne	mình biết nữa ne
14	1398	Disgust	mấy thằng củ lol việt nam nhảm nhí :))	mấy thằng củ lol việt_nam nhảm nhí
15	497	Sadness	tui thì ´ch va ¨i lã ´mày ma `ăn nhỉ `u n...	tui thì ´ch va_¨i lã ´mày ma `ăn_nhỉ `_u n...
16	562	Other	bếp dầu , nhiều nhà vẫn dùng	bếp dầu nhiều nhà vẫn dùng
17	327	Other	nếu thấy phụ nữ quá phức tạp để hiểu và chinh ...	nếu thấy phụ_nữ quá phức_tạp để hiểu và chinh_...
18	191	Fear	thời buổi bây chừ chuyện gì cũng có thể xảy ra...	thời_buổi bây_chừ chuyện gì cũng có_thể xảy ra...

Hình 3.2: Tập dữ liệu sau khi làm sạch

3.3 Chuẩn hóa dữ liệu đầu vào: Vector hóa văn bản

Sau giai đoạn tiền xử lý, để biểu diễn dưới dạng vector, hai phương pháp word embedding và bags of words được sử dụng.

Bags of words kết hợp với TF – IDF được sử dụng đối với hai mô hình học máy SVM và Random Forest.Hai mô hình LSTM và CNN, với các từ đã được huấn luyện trước, Word2Vec và fastText được sử dụng làm kĩ thuật chính.

3.4 Huấn luyện mô hình

Toàn bộ quá trình chạy thực nghiệm được tiến hành trên cấu hành máy và IDE có cấu hình như sau:

- Mã máy: Laptop Dell Inspiron 7559
- CPU: CPU Intel Core i7
- SSD: 120GB
- RAM 8GB
- Ngôn ngữ : Python phiên bản 3.9.6

- Thực thi: https://colab.research.google.com/drive

3.4.1 Mô hình Random Forest

Random Forest là một trong những thuật toán học máy hiệu quả nhất ngày nay. Thử nghiệm dựa trên thuật toán Random Forest cho số lượng cây quyết định là 256 và độ sâu của cây là 64.

3.4.2 Mô hình SVM

SVM là một thuật toán hiệu quả cho các bài toán phân loại với. Mô hình SVM sử dụng trong đề này này được hỗ trợ bởi thư viện scikit-learn.

Với mô hình học máy SVM và RF, phương pháp Grid-search được sử dụng để tìm kiếm tham số tốt nhất cho model.

SVM sử dụng bag of char (1, 7) và hinge loss function. Và để giảm việc overfitting quá mức, L2-regularization được thêm vào.

3.4.3 Mô hình LSTM

Với mô hình LSTM, sử dụng kiến trúc many-to-one do yêu cầu phân loại của bài toán. Lựa chọn các thông số phù hợp cho cảm xúc nhận dạng bằng tiếng Việt bằng cách thêm hai lớp Dropout lần lượt là 0,75 và 0,5 để tăng thời gian xử lý cũng như tránh tình trạng quá khớp (overfitting).

3.4.4 Mô hình CNN

Mô hình deep learning CNN được áp dụng 1 kernel chính với số filter là 128. Ngoài ra, Dropout là 0,2 và L2-regularization là 0,01 được áp dụng để tránh quá khớp.

3.5 Đánh giá mô hình

Mô hình trên sử dụng một số độ đo để đánh giá mô hình như: Learning curve, Classification report, Confusion matrix.

Learning curve: là đồ thị biểu diễn mối quan hệ giữa kinh nghiệm hoặc thời gian học tập và hiệu suất hoặc kỹ năng đạt được trong quá trình học. Đồ thị này thường có dạng

đường cong, cho thấy rằng ban đầu việc học có thể khó khăn và tiến bộ chậm, nhưng sau đó, khi kỹ năng và kiến thức tăng lên, việc học sẽ trở nên dễ dàng hơn và hiệu suất cải thiện nhanh chóng.

Classification report: là một bản tóm tắt hiệu suất của mô hình phân loại trong học máy. Báo cáo này cung cấp các chỉ số chính như precision, recall, F1-score cho từng lớp trong tập dữ liệu. Nó giúp đánh giá mức độ hiệu quả của mô hình trong việc phân loại các nhãn khác nhau.

Confusion matrix: là một công cụ đánh giá hiệu suất của mô hình phân loại trong học máy. Nó là một bảng thể hiện số lượng dự đoán đúng và sai của mô hình cho từng lớp. Ma trận này có các hàng đại diện cho các lớp thực tế và các cột đại diện cho các lớp dự đoán. Các ô trong ma trận cho biết số lần các mẫu thuộc một lớp thực tế cụ thể được mô hình dự đoán là một lớp khác. Nó giúp nhận diện các loại lỗi mà mô hình mắc phải.

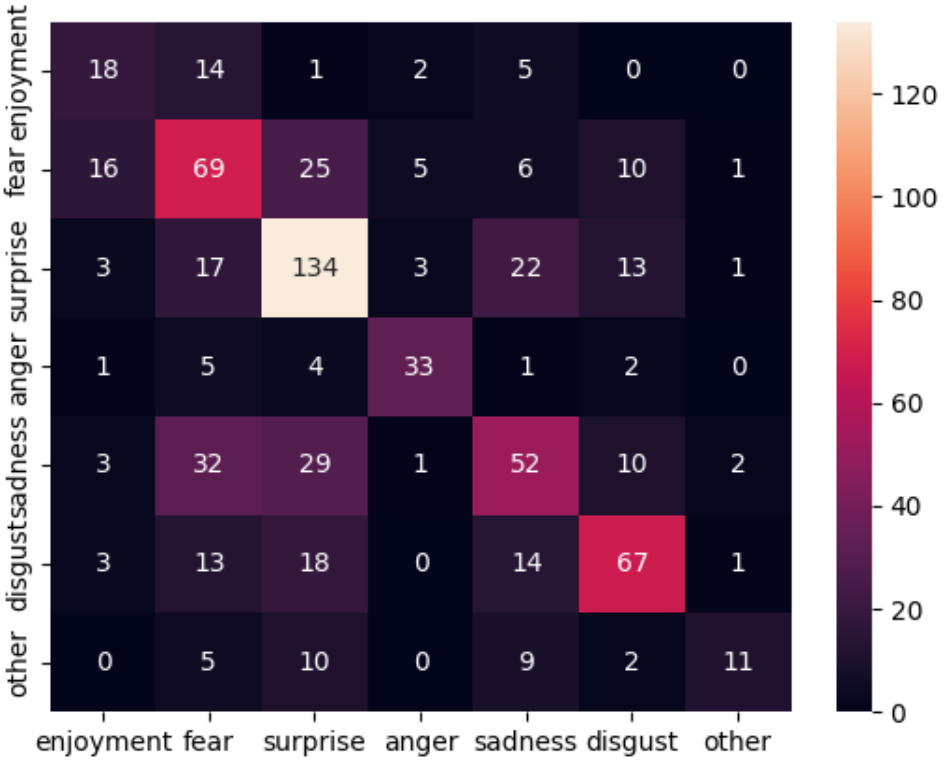
Chương 4 Kết quả

Để chứng minh hiệu suất của các mô hình phân loại, chúng tôi sử dụng confusion matrix để trực quan hóa sự tương quan giữa nhãn thực tế và nhãn dự đoán. Hình là confusion matrix của mô hình phân loại tốt nhất (SVM + TF IDF) trên kho ngữ liệu UIT-VSMEC. Sau đây là các kết quả chúng tôi thực nghiệm.

Classification Report:

	precision	recall	f1-score	support
Anger	0.41	0.45	0.43	40
Disgust	0.45	0.52	0.48	132
Enjoyment	0.61	0.69	0.65	193
Fear	0.75	0.72	0.73	46
Other	0.48	0.40	0.44	129
Sadness	0.64	0.58	0.61	116
Surprise	0.69	0.30	0.42	37
accuracy			0.55	693
macro avg	0.57	0.52	0.54	693
weighted avg	0.56	0.55	0.55	693

Hình 4.1: Classification report SVM



Hình 4.2: Confusion Matrix SVM

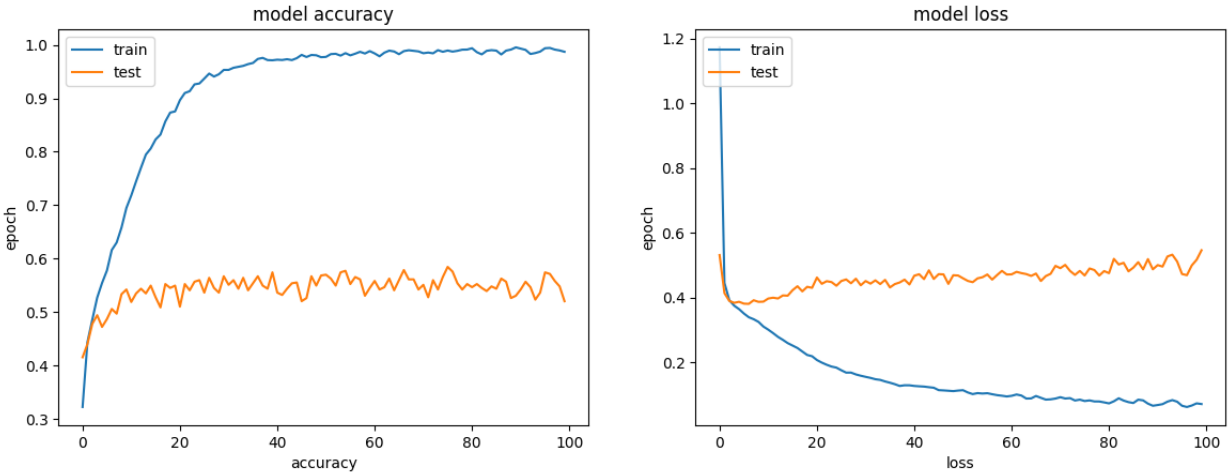
```
Best Parameters: {'rf__max_depth': 64, 'rf__n_estimators': 256}
Test Accuracy: 0.5093795093795094
Classification Report:
              precision    recall  f1-score   support

       Anger       0.50      0.05      0.09        40
     Disgust       0.54      0.55      0.54       132
   Enjoyment       0.46      0.82      0.59       193
        Fear       0.86      0.41      0.56        46
       Other       0.43      0.39      0.41       129
     Sadness       0.73      0.41      0.52       116
    Surprise       0.75      0.08      0.15        37

    accuracy                           0.51       693
   macro avg       0.61      0.39      0.41       693
weighted avg       0.56      0.51      0.48       693
```

Hình 4.3: Classification report Random Forest

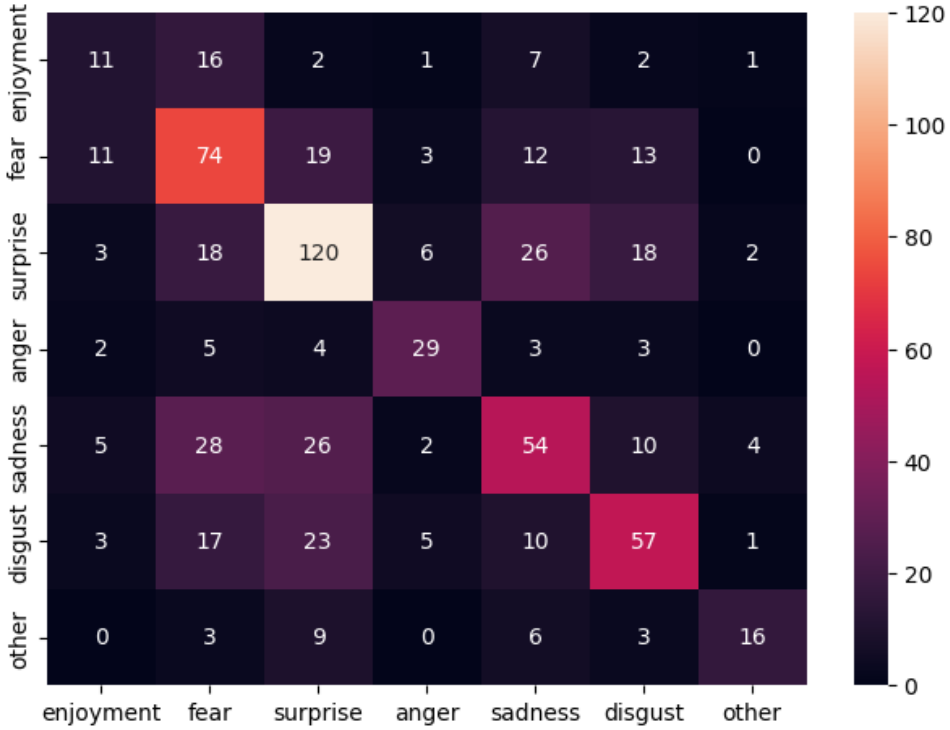
Hình 4.4: Confusion Matrix Random Forest

	precision	recall	f1-score	support
0	0.31	0.28	0.29	40
1	0.46	0.56	0.51	132
2	0.59	0.62	0.61	193
3	0.63	0.63	0.63	46
4	0.46	0.42	0.44	129
5	0.54	0.49	0.51	116
6	0.67	0.43	0.52	37
accuracy			0.52	693
macro avg	0.52	0.49	0.50	693
weighted avg	0.52	0.52	0.52	693

Hình 4.5: Classification report CNN sử dụng Word2Vec



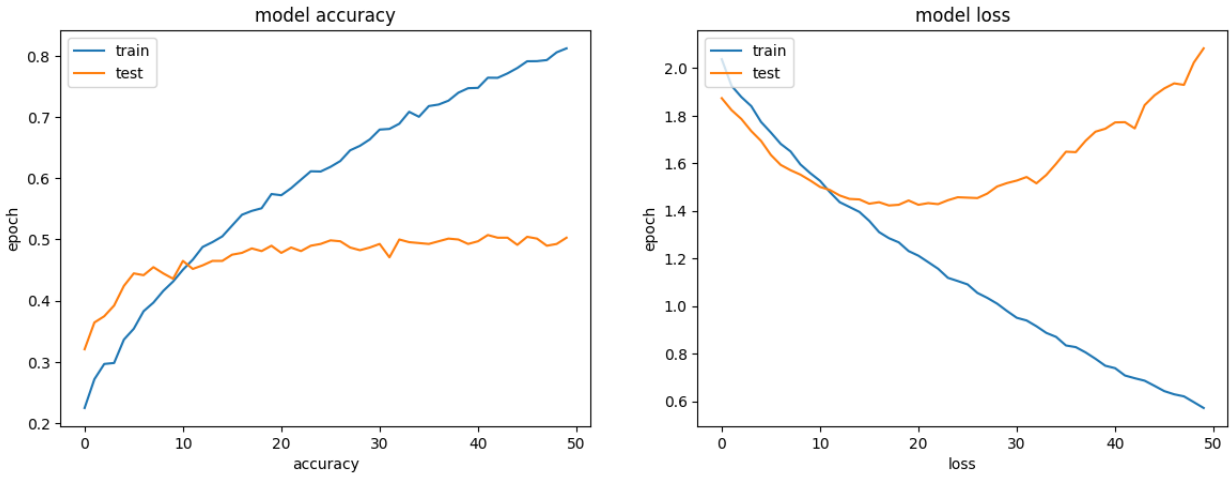
Hình 4.6: Learning curve CNN sử dụng Word2Vec



Hình 4.7: Confusion Matrix CNN sử dụng Word2Vec

	precision	recall	f1-score	support
0	0.53	0.25	0.34	40
1	0.49	0.53	0.51	132
2	0.57	0.61	0.59	193
3	0.74	0.67	0.70	46
4	0.45	0.51	0.48	129
5	0.54	0.53	0.54	116
6	0.63	0.32	0.43	37
accuracy			0.53	693
macro avg	0.56	0.49	0.51	693
weighted avg	0.54	0.53	0.53	693

Hình 4.8: Classification report LSTM sử dụng Word2Vec

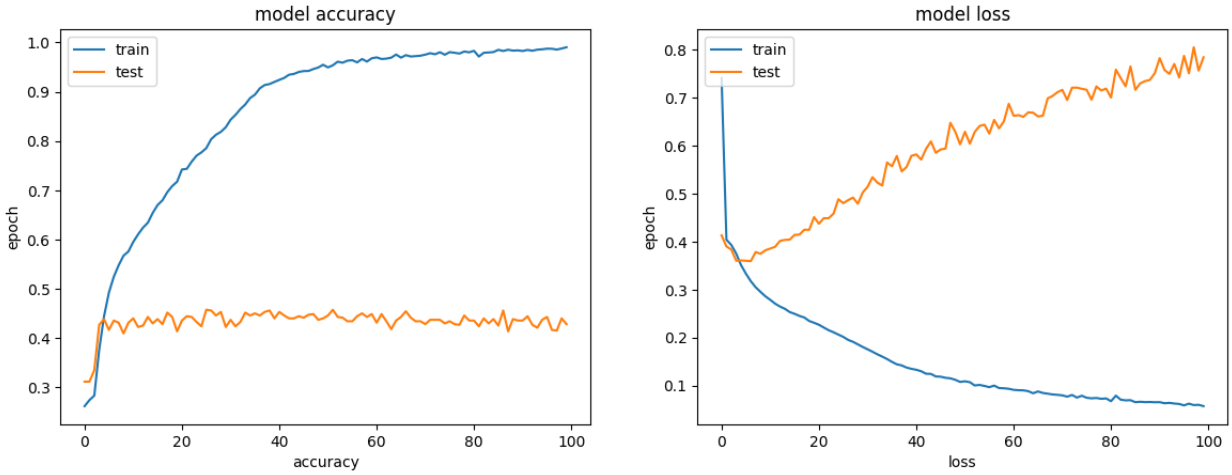


Hình 4.9: Learning curve LSTM sử dụng Word2Vec

Hình 4.10: Confusion Matrix LSTM sử dụng Word2Vec

	precision	recall	f1-score	support
0	0.23	0.30	0.26	40
1	0.50	0.55	0.52	132
2	0.57	0.52	0.55	193
3	0.52	0.57	0.54	46
4	0.40	0.44	0.42	129
5	0.51	0.45	0.48	116
6	0.50	0.32	0.39	37
accuracy			0.48	693
macro avg	0.46	0.45	0.45	693
weighted avg	0.49	0.48	0.48	693

Hình 4.11: Classification report CNN sử dụng fastText

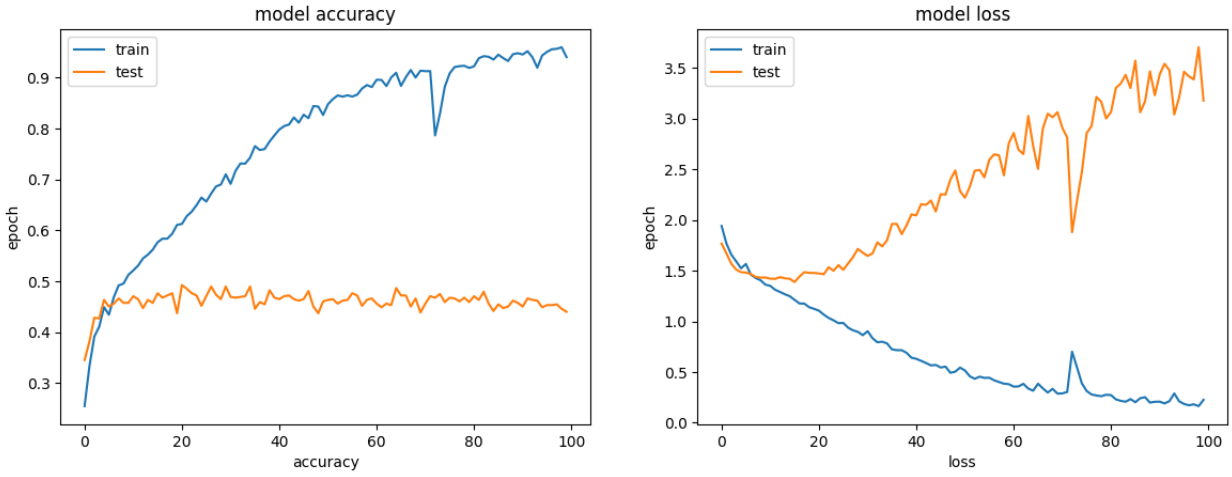


Hình 4.12: Learning curve CNN sử dụng fastText

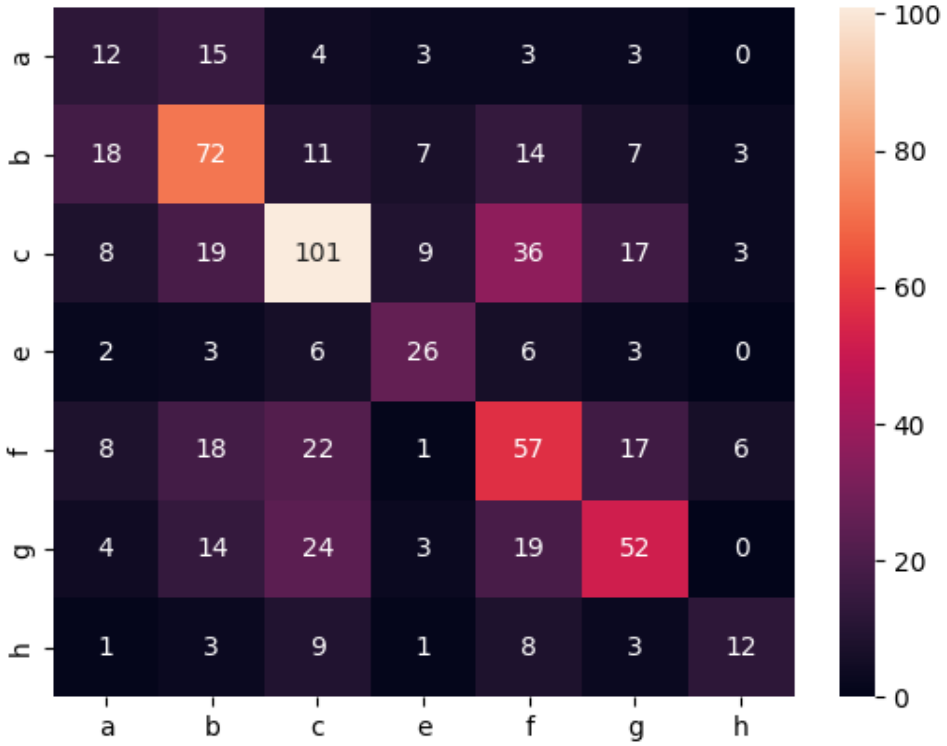
Hình 4.13: Confusion Matrix CNN sử dụng fastText

	precision	recall	f1-score	support
0	0.23	0.30	0.26	40
1	0.50	0.55	0.52	132
2	0.57	0.52	0.55	193
3	0.52	0.57	0.54	46
4	0.40	0.44	0.42	129
5	0.51	0.45	0.48	116
6	0.50	0.32	0.39	37
accuracy			0.48	693
macro avg	0.46	0.45	0.45	693
weighted avg	0.49	0.48	0.48	693

Hình 4.14: Classification report LSTM sử dụng fastText



Hình 4.15: Learning curve LSTM sử dụng fastText



Hình 4.16: Confusion Matrix LSTM sử dụng fastText

Chương 5 Kết luận

Từ phần kết quả thực nghiệm trên, chúng tôi rút ra kết luận chưa mô hình nào hoạt động tốt. Mô hình hoạt động tốt nhất trong số chúng tôi đang sử dụng là SVM với độ chính xác accuracy là 0.55. Nguyên nhân có thể do lượng dữ liệu đầu vào chưa đủ dẫn đến hiện tượng overfitting, cũng có thể do phương pháp tiền xử lý chưa đúng đắn và xây dựng mô hình chưa tốt. Đây là lĩnh vực mới, các thành viên trong nhóm tiếp cận với mục đích tìm hiểu thêm về một nhánh của ngành Trí tuệ nhân tạo - Xử lý ngôn ngữ tự nhiên, do đó kiến thức chuyên môn còn hạn chế.

Trong tương lai, nhóm muốn cải thiện số lượng cũng như chất lượng của kho ngữ liệu do hạn chế của những bình luận thể hiện cảm xúc tức giận, sợ hãi và ngạc nhiên. Ngoài ra mong muốn tiến hành thử nghiệm bằng cách sử dụng các mô hình học máy khác với các tính năng đặc biệt cũng như các mô hình học sâu với nhiều cách biểu diễn từ khác nhau hoặc kết hợp cả hai phương pháp trên kho ngữ liệu này.

Tài liệu tham khảo

[1] Quỳnh Nhật Phương Nguyễn and Xuân Thiện Bùi. “Nghiên cứu và xây dựng ứng dụng nhận diện cảm xúc của các bình luận trên mạng xã hội”. In: (2022).

[2] Paul Ekman. “Facial expression and emotion.”. In: *American psychologist* 48.4 (1993), p. 384.

[3] Vong Anh Ho, Duong Huynh-Cong Nguyen, Danh Hoang Nguyen, et al. “Emotion recognition for vietnamese social media text”. In: (2020), pp. 319–333.

[4] Y Kim. “Convolutional neural networks for sentence classification. arXiv [J]”. In: *preprint* (2014).