ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
——————— * ———————

BÁO CÁO CUỐI KỲ
MÔN: KHAI PHÁ WEB

Khai phá quan điểm:
Phân tích cảm xúc trên bình luận phim

Nhóm sinh viên thực hiện: Nhóm 06

STT	MSSV	Họ tên
1	20194185	Trịnh Đức Tiệp
2	20194102	Trần Thành Long
3	20194139	Trần Văn Phúc
4	20194182	Chu Mạnh Tiến

Mã lớp: 141351
GVHD: TS. Nguyễn Kiêm Hiếu

Hà Nội, tháng 7 năm 2023

MỤC LỤC

PHÂN CHIA CÔNG VIỆC

MSSV	Họ tên	Công việc
20194185	Trịnh Đức Tiệp	Triển khai hai mô hình Multinomial Naive Bayes trên BOW và LSTM trên mô hình nhúng Word2Vec
20194102	Trần Thành Long	Thu thập dữ liệu, tổng hợp và làm sạch dữ liệu, làm báo cáo và slide thuyết trình
20194139	Trần Văn Phúc	Triển khai hai mô hình Gaussian Naive Bayes trên mô hình nhúng Word2Vec và LSTM trên Tokenizer
20194182	Chu Mạnh Tiến	Thu thập dữ liệu, tiền xử lý dữ liệu, thực hiện nhúng từ Word2Vec

1. Mô tả bài toán

1.1. Tên bài toán

Phân tích cảm xúc trên bình luận phim

1.2. Mô tả bài toán

1.2.1. Giới thiệu bài toán

Bài toán "phân tích cảm xúc trên bình luận phim" là một trong những bài toán quan trọng của lĩnh vực xử lý ngôn ngữ tự nhiên. Mục đích của bài toán là phân loại một bình luận phim thành ba nhóm: tích cực (positive), tiêu cực (negative) hoặc trung tính (neutral). Tác dụng của bài toán là giúp các nhà sản xuất phim, các nhà phê bình phim, các trang web đánh giá phim,... đánh giá được cảm nhận của khán giả về một bộ phim, từ đó có thể cải thiện sản phẩm và tăng doanh thu.

Đồng thời, bài toán phân tích cảm xúc cũng có ý nghĩa trong nghiên cứu về xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo, đặc biệt là trong lĩnh vực học sâu (deep learning). Bài toán cũng là một trong những bài toán tiêu biểu để áp dụng các phương pháp học máy như mạng nơ-ron, học sâu, và các phương pháp xử lý ngôn ngữ tự nhiên.

Tóm lại, bài toán phân tích cảm xúc trên bình luận phim có tính ứng dụng cao và có ý nghĩa quan trọng trong nhiều lĩnh vực.

1.2.2. Phân tích bài toán

Sau đây ta sẽ phân tích input và output của bài toán.

Input: Tập các văn bản là các bình luận, đánh giá cho một bộ phim

Output: Nhãn quan điểm cho các bình luận theo 3 mức:

STT	Nhãn	Ý nghĩa
1	Negative	Tiêu cực
2	Neutral	Trung tính
3	Positive	Tích cực

Bài toán phức tạp do cảm xúc trong các bình luận phim có thể mang tính chất đa dạng và thay đổi. Một số bình luận có thể rất rõ ràng và mạnh mẽ trong việc diễn đạt cảm xúc, trong khi những bình luận khác có thể mang tính chất mập mờ hoặc không rõ ràng. Điều này đòi hỏi mô hình phải có khả năng nhạy bén và linh hoạt để hiểu và phân tích được các cấu trúc ngôn ngữ, từ ngữ, biểu đạt cảm xúc, ngữ cảnh và ý nghĩa của bình luận.

Một số thách thức khác trong bài toán này là việc xử lý các biến thể ngôn ngữ và ngữ cảnh khác nhau. Những ngôn ngữ không chuẩn, từ viết tắt, ngôn ngữ nhập nhằng và văn phong đa dạng có thể xuất hiện trong các bình

luận phim. Mô hình phân tích cảm xúc phải có khả năng hiểu và xử lý các đặc điểm này để đưa ra kết quả phân loại chính xác.

2. Tập dữ liệu và các phương pháp, mô hình đề xuất

2.1. Tập dữ liệu

Tập dữ liệu cho hai nhãn lớp positive và negative sử dụng được thu thập từ nguồn: www.kaggle.com

+ Tên tập dữ liệu: *IMDB Dataset of 50K Movie Reviews*
+ Link: Link tập dữ liệu
+ Số lượng mẫu: 49.582 bình luận

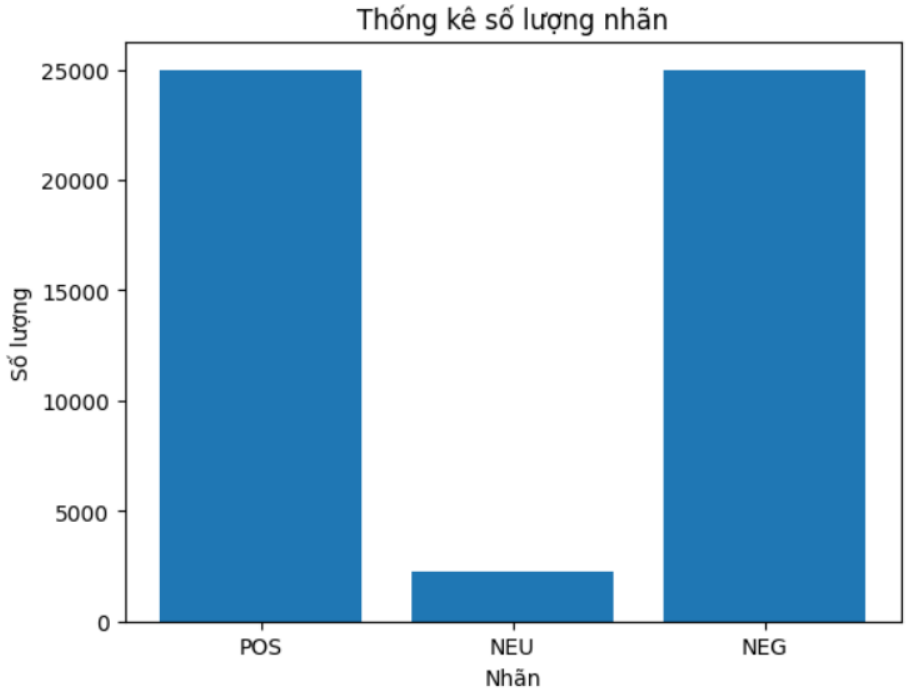
Tập dữ liệu cho nhãn neutral được trích suất từ nguồn: github.com

+ Tên tập dữ liệu: Sentiment-Analysis-of-Movie-review-dataset
+ Link: Link tập dữ liệu trung tính
+ Số lượng mẫu: 2.500 bình luận trung tính

+ Một bình luận mẫu:

"I sure would like to see a resurrection of a up dated Seahunt series with the tech they have today it would bring back the kid excitement in me.I grew up on black and white TV and Seahunt with Gunsmoke were my hero's every week.You have my vote for a comeback of a new sea hunt.We need a change of pace in TV and this would work for a world of under water adventure.Oh by the way thank you for an outlet like this to view many viewpoints about TV and the many movies.So any ole way I believe I've got what I wanna say.Would be nice to read some more plus points about sea hunt.If my rhymes would be 10 lines would you let me submit,or leave me out to be in doubt and have me to quit,If this is so then I must go so lets do it."

Số lượng bình luận có nhãn trung tính là khá ít so với các bình luận có nhãn tích cực và tiêu cực:



2.2. Đề xuất Phương pháp thực hiện

Bước 1. Tiền xử lý dữ liệu

Sau khi thu thập được dữ liệu, chúng ta cần tiền xử lý để chuẩn hóa và làm sạch dữ liệu. Các bước tiền xử lý: lọc bình luận nhiễu, định dạng lại dữ liệu, gộp các file dữ liệu lại với nhau, làm sạch dữ liệu với các bước: loại bỏ thẻ html, từ dừng, sửa các từ viết tắt, …

Bước 2. Nhúng từ

Trích xuất đặc trưng (Feature extraction): Để phân loại cảm xúc trên bình luận phim, chúng ta cần biểu diễn các bình luận dưới dạng các đặc trưng có thể được sử dụng bởi mô hình học máy. Ở đây, nhóm đã lựa chọn mô hình Word2Vec với phương pháp skip-gram.

Bước 3. Xây dựng mô hình

Chia tập dữ liệu đã được nhúng từ thành 2 tập train và test

Nhóm lựa chọn xây dựng theo 2 mô hình Naive Bayes (với MultiNomial & BOW và Gaussian & word2vec) và LSTM (với word2vec và tokenizer) để thực hiện huấn luyện.

Bước 4. Đánh giá mô hình

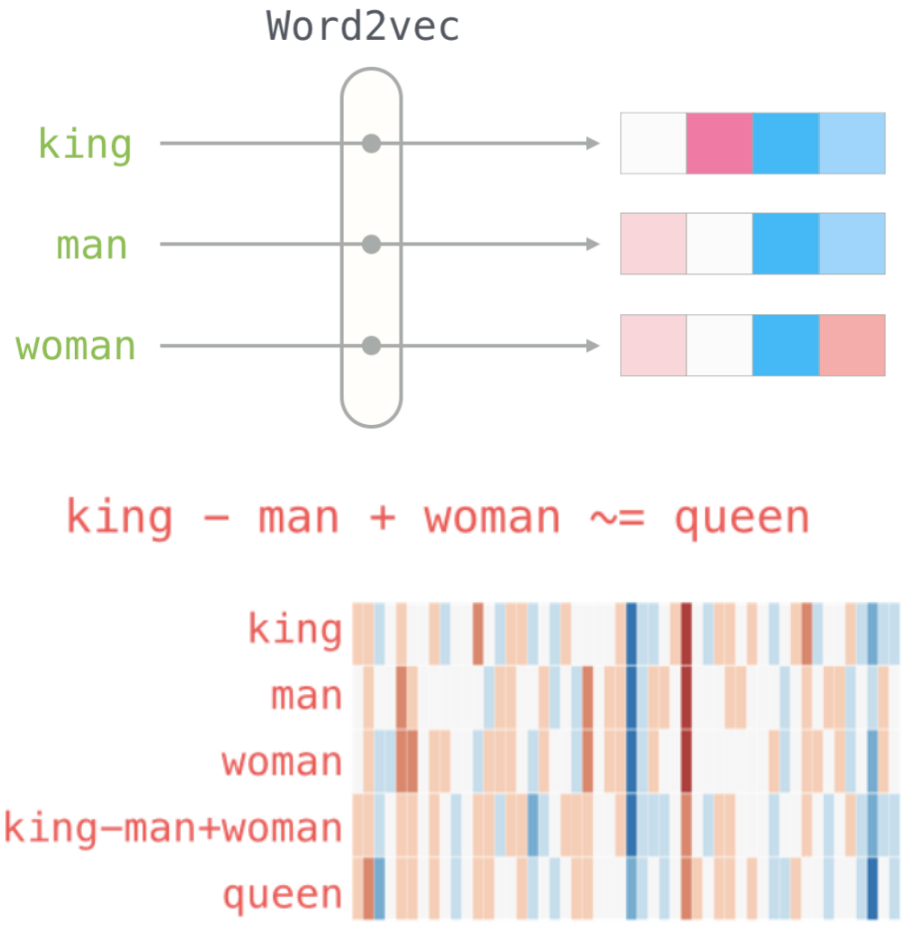
Thực hiện phân loại cảm xúc trên tập test

Sau đó thực hiện tính toán các độ đo trên hai mô hình và đưa ra đánh giá, đồng thời so sánh hai mô hình trên bài toán này

3. Cơ sở lý thuyết

3.1. Phương pháp nhúng từ Word2Vec

Word2Vec là một phương pháp nhúng từ (word embedding) tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên và học máy. Nó giúp biểu diễn từng từ trong không gian đa chiều dưới dạng các vector số học có khả năng biểu thị mối quan hệ ngữ nghĩa và cú pháp giữa các từ. Các biểu diễn từ trong Word2Vec cung cấp không chỉ thông tin về đồng nghĩa và trái nghĩa của từ, mà còn về mối quan hệ cú pháp giữa chúng. Các phép toán vector có thể được thực hiện để tìm từ gần nhất, tính toán sự tương đồng cosine giữa các từ, và thậm chí thực hiện các phép tính toán văn bản như:

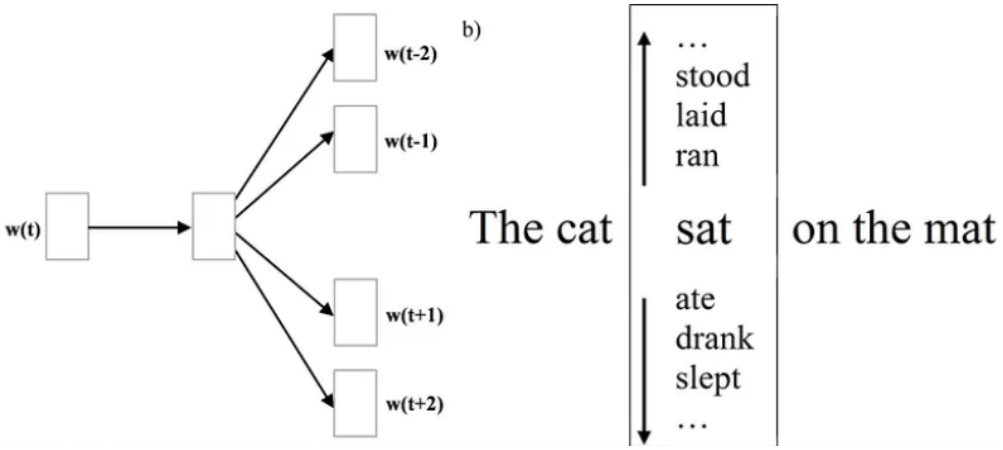


Trong Word2Vec, có hai phương pháp chính được sử dụng để tạo ra các nhúng từ:

+ Continuous Bag-of-Words (CBOW): được sử dụng để dự đoán từ hiện tại dựa trên ngữ cảnh xung quanh nó. Cụ thể, nó cố gắng dự đoán từ hiện tại bằng cách sử dụng các từ xung quanh nó trong một cửa sổ ngữ cảnh cố định.

+ Skip-gram: Đây là một phương pháp khác trong Word2Vec, ngược lại với CBOW. Trong Skip-gram, mục tiêu là dự đoán các từ xung quanh từ hiện tại dựa trên từ hiện tại.

Trong phần này, chúng ta sẽ tìm hiểu về phương pháp nhúng từ Skip-gram. Quá trình huấn luyện Skip-gram bắt đầu bằng việc tạo ra các cặp từ và ngữ cảnh trong văn bản. Cặp từ bao gồm một từ hiện tại và một từ ngữ cảnh trong khoảng cách xác định. Mô hình sử dụng một lớp ẩn có kích thước cố định để biểu diễn các từ trong không gian vector. Mỗi từ trong từ vựng được biểu diễn bằng một vector số học dựa trên các trọng số của lớp ẩn. Quá trình huấn luyện của mô hình nhằm điều chỉnh các trọng số này để tối thiểu hóa sai số giữa dự đoán và ngữ cảnh thực tế.



Ví dụ: Từ “sat” sẽ được đưa ra và ta sẽ cố gắng dự đoán các từ “cat”, “mat” ở vị trí -1 và 3 tương ứng với “sat” ở vị trí 0 (không dự đoán các từ dừng: on, the).

Sau quá trình huấn luyện, mô hình Skip-gram sẽ tạo ra các biểu diễn vector cho từng từ trong từ vựng. Các vector này có thể được sử dụng để đo đạc sự tương đồng ngữ nghĩa hoặc cú pháp giữa các từ. Mô hình Skip-gram có thể được huấn luyện trên các tập dữ liệu lớn và tạo ra các nhúng từ chất lượng cao. Khi một từ mới xuất hiện, ta có thể sử dụng mô hình đã được huấn luyện để lấy biểu diễn vector của từ đó.

Phương pháp nhúng từ Word2Vec, đặc biệt là Skip-gram, đã chứng tỏ hiệu quả và sức mạnh của nó trong nhiều ứng dụng xử lý ngôn ngữ tự nhiên. Từ việc cải thiện hiểu biết ngôn ngữ tự nhiên cho đến ứng dụng trong học máy và khai thác thông tin, Word2Vec đã đóng góp đáng kể cho sự phát triển của lĩnh vực này.

3.2. Mô hình phân loại theo xác suất Naive Bayes

Naive Bayes Classification (NBC) là một thuật toán phân loại dựa trên tính toán xác suất áp dụng định lý Bayes. Thuật toán này thuộc nhóm Supervised Learning (Học có giám sát).

Theo định lý Bayes, ta có công thức tính xác suất như sau:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Do đó ta có:

$$P(y|x) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

Trên thực tế thì ít khi tìm được dữ liệu mà các thành phần là hoàn toàn độc lập với nhau. Tuy nhiên giả thiết này giúp cách tính toán trở nên đơn giản, training data nhanh, đem lại hiệu quả bất ngờ với các lớp bài toán nhất định.

Trong thuật toán Naive Bayes, nhóm sẽ sử dụng 2 mô hình:

+ Mô hình Multinomial Naive Bayes: được sử dụng trong phân loại văn bản mà feature vectors được tính bằng Bags of Words. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó. Ta tính xác suất từ xuất hiện trong văn bản P(xi | y) như sau:

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

Trong đó:

+ N_i là tổng số lần từ x_i xuất hiện trong văn bản.
+ N_c là tổng số lần từ của tất cả các từ $x_1,\ldots, x_n$ xuất hiện trong văn bản.

Công thức trên có hạn chế là khi từ xi không xuất hiện lần nào trong văn bản, ta sẽ có $N_i=0$. Điều này làm cho $P(x_i \mid y)=0$. Để khắc phục vấn đề này, người ta sử dụng kỹ thuật gọi là Laplace Smoothing bằng cách cộng thêm vào cả tử và mẫu để giá trị luôn khác 0:

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

Trong đó:

α thường là số dương, bằng 1.

dα được cộng vào mẫu để đảm bảo $\sum_{i=1}^{d} P(x_i|y) = 1$

+ Mô hình GaussianNB với Word2Vec Skip-gram đã được nhúng từ trước. Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành

phần là các biến liên tục. Với mỗi chiều dữ liệu i và một class c, xi tuân theo một phân phối chuẩn có kỳ vọng μ_{ci} và phương sai σ_{ci}^2:

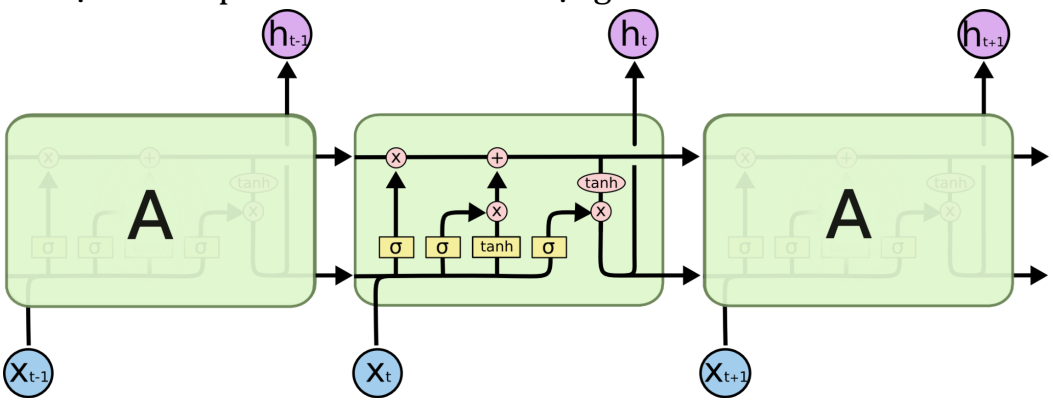
$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right)$$

Trong đó, bộ tham số $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ được xác định bằng Maximum Likelihood:

$$(\mu_{ci}, \sigma_{ci}^2) = \arg\max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^{N} p(x_i^{(n)}|\mu_{ci}, \sigma_{ci}^2)$$

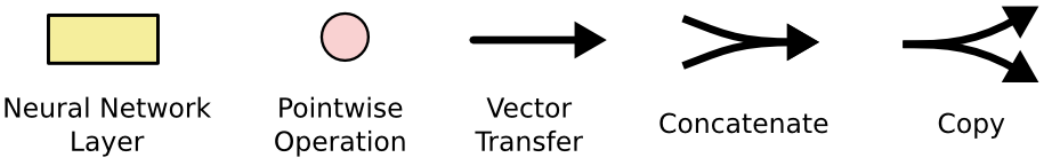
3.3. Mô hình Long-short term memory

Long short-term memory (LSTM) là một kiến trúc artificial recurrent neural network (RNN), LSTM có các kết nối phản hồi, nó có thể xử lý không chỉ các điểm dữ liệu đơn lẻ (chẳng hạn như hình ảnh) mà còn toàn bộ chuỗi dữ liệu (chẳng hạn như speech hoặc video). LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.



Sự lặp lại kiến trúc module trong mạng LSTM chứa 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác

Các kí hiệu có thể diễn giải như sau:



Trong sơ đồ tính toán trên, mỗi một phép tính sẽ triển khai trên một vector. Trong đó hình tròn màu hồng biểu diễn một toán tử đối với vector như phép cộng vector, phép nhân vô hướng các vector. Màu vàng thể hiện hàm activation mà mạng nơ ron sử dụng để học trong tầng ẩn, thông thường là các hàm phi tuyến sigmoid và tanh. Kí hiệu 2 đường thẳng nhập vào thể hiện phép chập kết quả trong khi kí hiệu 2 đường thẳng rẽ nhánh thể hiện cho nội dung vector trước đó được sao chép để đi tới một phần khác của mạng nơ ron.

4. Các bước thực hiện

4.1. Tiền xử lý dữ liệu

Các bước thực hiện:

+ Loại bỏ các phần tử thẻ html khỏi các review: Các phần tử thẻ HTML không mang ý nghĩa trong việc phân tích cảm xúc và được loại bỏ để tập trung vào nội dung chính của đánh giá.

Sử dụng công cụ thư viện regex:

```
for review in clone:
  # replace html tag to empty string: ''
  review['review'] = re.sub(r'<[^>]+>', '', review['review'])
```

+ Loại bỏ các ký tự biểu thị cảm xúc (emoji) trong các đánh giá: Các biểu tượng cảm xúc không cần thiết có thể ảnh hưởng đến quá trình phân tích cảm xúc và được loại bỏ khỏi văn bản.

```
for review in clone:
  review['review'] = review['review'].encode("ascii","ignore")
  review['review'] = review['review'].decode()
```

+ Loại bỏ các con số: Con số không mang ý nghĩa cụ thể trong việc phân tích cảm xúc và được loại bỏ để tập trung vào các từ và ngữ cảnh trong văn bản.

```
for review in clone:
  review['review'] = re.sub(r'\d', '', review['review'])
```

+ Loại bỏ dấu câu: Dấu câu thường không mang nhiều thông tin ý nghĩa và có thể gây nhiễu trong quá trình phân tích. Do đó, chúng được loại bỏ khỏi văn bản.

```
for review in clone:
  review['review'] = re.sub(r'[^\w\s]', ' ', review['review'])
```

+ Loại bỏ từ dừng (stop words): Từ dừng là các từ phổ biến trong ngôn ngữ (ví dụ: "và", "là", "được") không mang nhiều ý nghĩa đặc biệt và có thể loại bỏ để tập trung vào các từ quan trọng hơn trong văn bản.

Danh sách các từ dừng tiêu chuẩn của thư viện nltk. tiếng Anh được tải về từ thư viện nltk: STOPWORDS

```
for review in clone:
  review['review'] = ' '.join([word for word in review['review'].split() if word not in STOPWORDS])
```

+ Xóa bỏ đi các từ trong review có độ dài nhỏ hơn 2 vì đây thường là các từ bị gõ nhầm, gõ thừa hoặc bị sai chính tả và ít mang thông tin ý nghĩa trong đó.

```
for review in clone:
  review['review'] = ' '.join([word for word in review['review'].split() if len(word) > 2])
```

+ Khôi phục các từ viết tắt trong tiếng Anh. Một số từ được viết tắt có thể làm sai lệch ý nghĩa ban đầu của nó. Do đó, các từ viết tắt được chuyển đổi trở lại dạng ban đầu để đảm bảo ý nghĩa chính xác.

Sử dụng thư viện contractions:

```
for review in clone:
  review['review'] = contractions.fix(review['review'])
```

+ Stemming hoặc Lemmatizing các từ: nghĩa là giảm từ về dạng gốc để tạo sự đồng nhất giữa các từ có cùng nguồn gốc (có thể xem xét ngữ cảnh, ngữ nghĩa hoặc không). Ví dụ, từ "running", "runs", "runners" sẽ được cắt bỏ các hậu tố để trở thành từ gốc "run". Điều này giúp tạo sự thống nhất và giảm số lượng biến thể của các từ trong văn bản, từ đó giúp quá trình phân tích cảm xúc hiệu quả hơn.

Sử dụng công cụ stem thư viện nltk

```
[ ]  from nltk.stem import WordNetLemmatizer
     lemmatizer = WordNetLemmatizer()
     # nltk.download('wordnet')

[ ]  for review in clone:
       review['review'] = lemmatizer.lemmatize(review['review'])
```

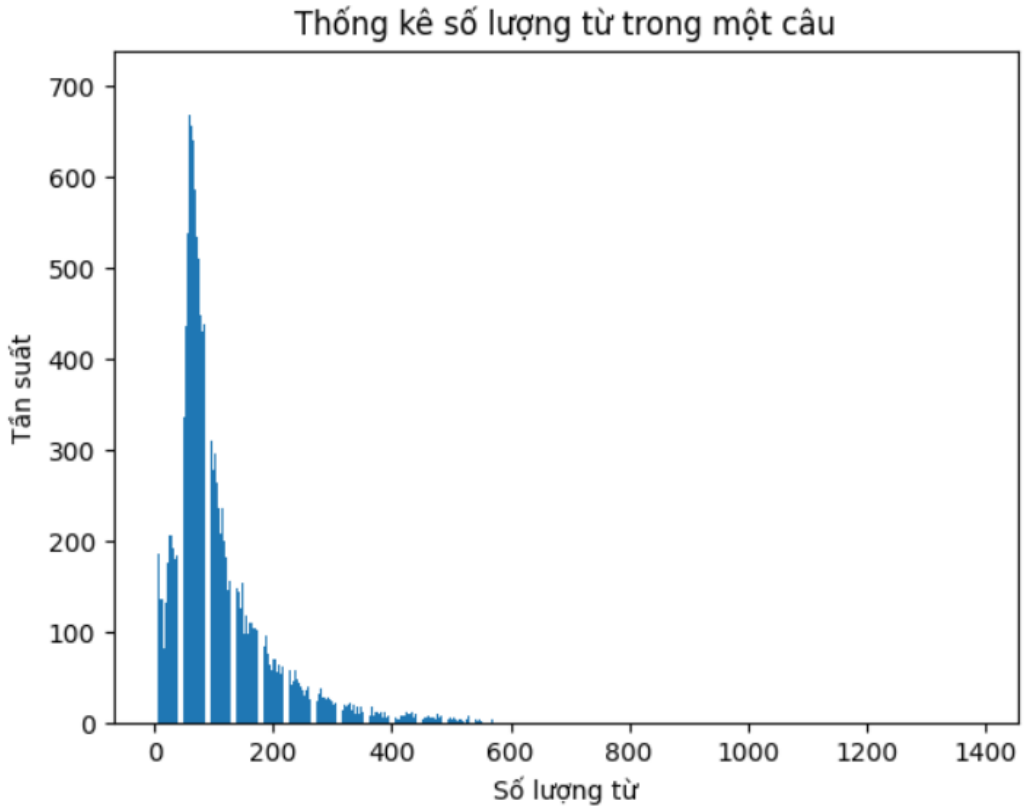
4.2. Nhúng từ (Word Embedding) bằng Word2Vec

Có nhiều phương pháp có thể dùng để nhúng từ như PMI, Word2Vec, GloVe hoặc thậm chí là làm thủ công. Trong đề tài này, nhóm sẽ lựa chọn công cụ Word2Vec để nhúng các từ trong mỗi bình luận. Word2Vec có hai phương pháp chính là CBOW và Skipgram. Ở đây nhóm cũng chỉ ra rõ rằng phương pháp sẽ được sử dụng là Skipgram.

Các bước thực hiện như sau:

+ Chia các câu thành các từ (tokens) riêng lẻ, kết quả của quá trình này sẽ được lưu trong cột dữ liệu mới. Mỗi phần tử trong cột này sẽ là một danh sách các từ đã được tách ra từ câu tương ứng.

Kết quả sau khi chia câu thành các từ:



+ Xây dựng mô hình skip-gram với tập dữ liệu trên.

Sử dụng thư viện hỗ trợ gensim

```
# Xây dựng mô hình Word2Vec với mô hình Skip-gram
skip_gram = Word2Vec(
    sentences     = data['tokens'],
    vector_size   = 100,  # kích thước vector từ nhúng (thường 100-300)
    window        = 5,    # kích thước cửa sổ quét
    min_count     = 1,    # số lần xuất hiện tối thiểu một từ được nhúng
    sg            = 1     # lựa chọn mô hình skip-gram
)
```

Một số tham số của mô hình:

+ sentences: tập từ vựng, được lấy từ các tokens của các câu sau khi chia tách ở bước trên

+ vector_size: Kích thước của vector từ sau khi nhúng. Nhóm chọn giá trị cho trường này là 100.

+ window: kích thước cửa sổ quét khi nhúng từ. Giá trị lựa chọn là 5.

+ min_count: Những từ được nhúng là những từ xuất hiện tối thiểu min_count lần. Ở đây nhóm em cho phép mỗi từ chỉ cần xuất hiện một lần là được nhúng vào mô hình.

+ sg: cờ báo hiệu loại mô hình W2V là Skip-gram hay CBOW, ở đây sg = 1 cho thấy mô hình được nhúng là Skip-gram

4.3. Xây dựng mô hình Naive Bayes

4.3.1. Naive Bayes với Bag of Words

+ Chia dữ liệu thành 2 tập train và test tỷ lệ 80:20
+ Bag of word:
  Sử dụng CountVectorizer để xây dựng ma trận đặc trưng từ dữ liệu văn bản

```
cv=CountVectorizer(min_df=0,max_df=1,binary=False,ngram_range=(1,3))
# xây dựng tập từ điển từ data_train và đồng thời tạo ma trận đặc trưng dựa trên nó
cv_train=cv.fit_transform(data_train)
# tạo ma trận đặc trưng dựa trên tập từ điển đã tạo
cv_test=cv.transform(data_test)
```

  Một số tham số của mô hình:
  . min_df=0 chỉ định rằng từ xuất hiện ít nhất 0 lần trong tập huấn luyện sẽ được bao gồm trong từ điển. Tức là không có yêu cầu về mức tối thiểu xuất hiện của một từ.
  . max_df=1 chỉ định rằng từ xuất hiện tối đa 1 lần trong tập huấn luyện sẽ được bao gồm trong từ điển. Tức là không có yêu cầu về mức tối đa xuất hiện của một từ.
  . binary=False chỉ định rằng giá trị đếm của mỗi từ sẽ được sử dụng thay vì giá trị nhị phân (0 hoặc 1). Điều này cho phép chúng ta lưu giữ thông tin về tần suất xuất hiện của các từ.
  . ngram_range=(1,3) chỉ định rằng chúng ta muốn xem xét các từ đơn (1-gram) và các cụm từ liên tiếp gồm 2 từ (2-gram) và 3 từ (3-gram).

+ Số hoá và chia tập nhãn lớp: Các nhãn lớp trong tập huấn luyện và tập kiểm tra sẽ được số hoá để chuẩn bị cho quá trình huấn luyện và dự đoán. Kết quả sau khi số hoá negative: [1 0 0], neutral: [0 1 0], positive: [0 0 1]
+ Huấn luyện với Multinomial Naive Bayes với Bag of Word:
  Mô hình phân loại sử dụng phân phối đa thức để ước lượng xác suất của từng nhãn lớp

```
# mô hình huấn luyện
mnb=MultinomialNB()
# huấn luyện mô hình trên tập Bad of Word
mnb_bow=mnb.fit(cv_train,train_label_data)
```

  Ma trận đặc trưng từ tập huấn luyện (cv_train) và nhãn lớp tương ứng với từng văn bản trong tập huấn luyện (train_label_data).
+ Thực hiện dự đoán trên tập test

```
mnb_bow_predict=mnb.predict(cv_test)
```

4.3.2. Naive Bayes với Word2vec skip-gram

a) Số hóa input

Mô hình Naive Bayes yêu cầu dữ liệu vào phải là dạng mảng number hai chiều. Trong đó, chiều thứ nhất là số lượng mẫu (tương ứng với số lượng bình luận, số lượng sentence), chiều thứ hai là kích thước vector số của mỗi sentence. Vậy, để huấn luyện theo mô hình này, ta sẽ lấy mỗi câu sẽ là mean của các word trong câu ấy.

Nhãn lớp cũng cần số hóa hợp lý. Ở đây ta sẽ mã hóa nhãn lớp như sau: lớp positive mang nhãn 0, neutral nhãn 1, negative nhãn 2.

b) Tiến hành mô hình hóa

Chia dữ liệu thành tập huấn luyện và tập đánh giá tỷ lệ 80:20

Tạo mô hình GaussianNB và huấn luyện với tập train

Mô hình phân loại sử dụng phân phối Gaussian (phân phối chuẩn) để ước lượng xác suất của từng nhãn lớp.

```
gaussianNB = GaussianNB()
gaussianNB.fit(X_train, y_train)

joblib.dump(gaussianNB, 'gaussianNB.pkl')
```

+ Thực hiện dự đoán trên tập test

```
predict = gaussianNB.predict(X_test)
```

4.4. Xây dựng mô hình LSTM

4.4.1. LSTM với Word2Vec

+ Dữ liệu huấn luyện chính là mảng các câu, từ đã được nhúng bằng phương pháp w2v - Skipgram trong phần 4.2.
+ Số hóa nhãn: Các nhãn được số hóa theo onehot như sau:
    + Nhãn positive: [1, 0, 0]
    + Nhãn neutral: [0, 1, 0]
    + Nhãn negative: [0, 0, 1]
+ Padding lại các câu:

Dữ liệu huấn luyện trong LSTM yêu cầu phải là các câu có cùng số từ với nhau. Nhóm thực hiện padding lại các câu:

+ Mỗi câu nhóm lựa chọn sẽ có 150 từ
+ Với các câu có nhiều hơn 150 từ sẽ bỏ đi các từ cuối
+ Với các câu có ít hơn 150 từ sẽ thực hiện thêm các vector-không vào cuối mỗi câu để cho đủ 150 từ đem huấn luyện

Kích thước tập dữ liệu huấn luyện là 40.000 mẫu x 150 từ/câu x 100.

+ Chia tập dữ liệu theo tỷ lệ train : test = 80 : 20

4.4.2. LSTM với Tokenizer

+ Dữ liệu huấn luyện sẽ không lấy từ mảng các câu đã được w2v từ bước trước mà sẽ lấy trực tiếp từ data sau khi tiền xử lý.
+ Sử dụng tokenizer để đánh số hiệu các từ trong từ điển, một câu sẽ đưa về mảng số hiệu của từ xuất hiện trong câu ấy,
  Số lượng từ trong từ điển maximum = 25000 từ.
  Padding mỗi câu đưa về số lượng từ trong một câu bằng 250.
+ Số hóa nhãn: Các nhãn được số hóa theo onehot như sau:
    + Nhãn positive: [1, 0, 0]
    + Nhãn neutral: [0, 1, 0]
    + Nhãn negative: [0, 0, 1]
+ Dữ liệu chia theo tỉ lệ: train : test = 80 : 20
+ Nhúng câu sau khi số hóa bằng tầng Embedding:

Mô hình mạng neural khi train theo phương pháp này sẽ sử dụng tầng Embedding làm tầng đầu tiên, dùng để nhúng câu sau khi tokenize để làm input cho tầng LSTM.

```
    model.add(tf.keras.layers.Embedding(input_dim=max_numb
er_words,output_dim=128,input_length=X.shape[1]))
```

Tầng embedding này sẽ nhúng các từ và đưa chúng về vector có kích thước bằng 128.

4.4.3. Mô hình mạng neural huấn luyện

Mô hình mạng neural dùng để huấn luyện mà nhóm thiết kế có cấu trúc gồm các tầng:

STT	Tầng	Tham số
1	LSTM	Số neural = 128, hàm kích hoạt: tanh
2	SpatialDropout1D	drop out = 0.2
3	LSTM	Số neural = 64, hàm kích hoạt: tanh
4	Dense	Số neural = 3, hàm kích hoạt: softmax

Mạng được huấn luyện trên 10 epochs, kích thước batch_size = 225 và tỉ lệ validation_split=0.4, hàm lỗi: binary cross entropy.

5. Kết quả thực nghiệm

5.1. Kết quả huấn luyện bằng Naive Bayes

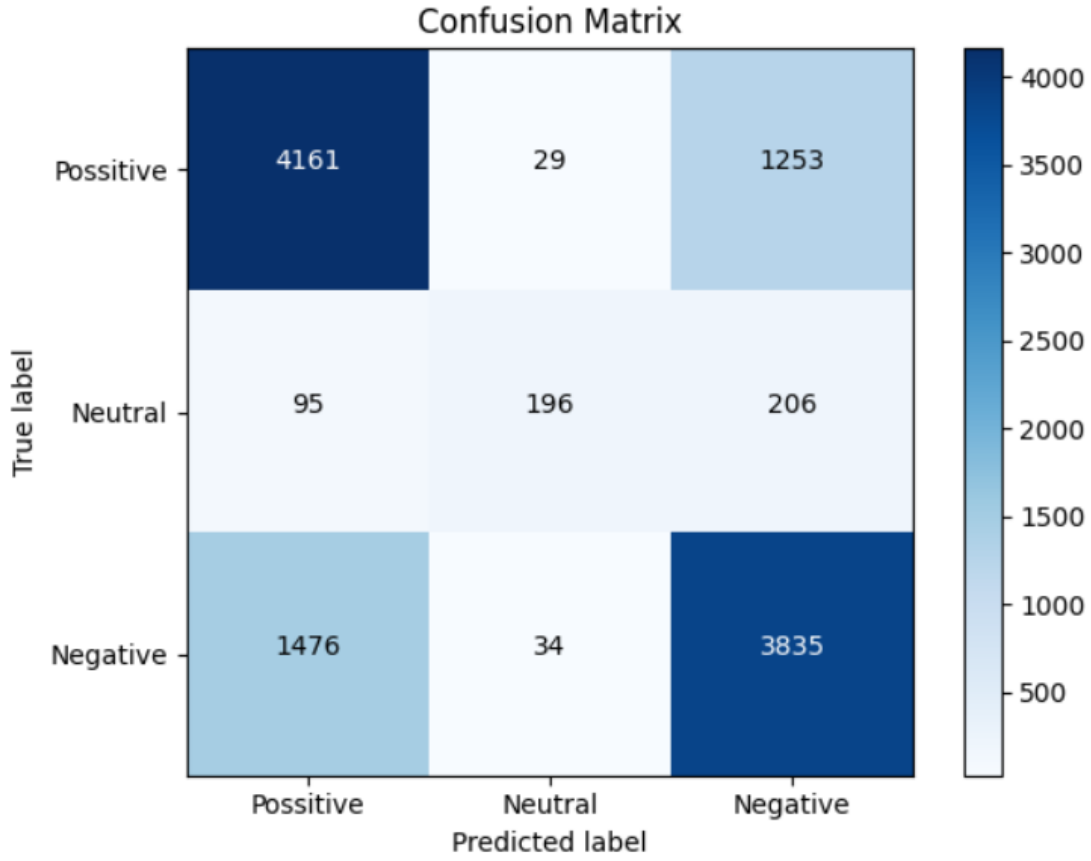
5.1.1. Naive bayes với MultiNomial và BOW:

- Số lượng nhãn dự đoán và nhãn thực tế:

. Predict: [5732 259 5294]

. Reality: [5443 497 5345]

- Ma trận nhầm lẫn:



- Giá trị Precision, Recall:

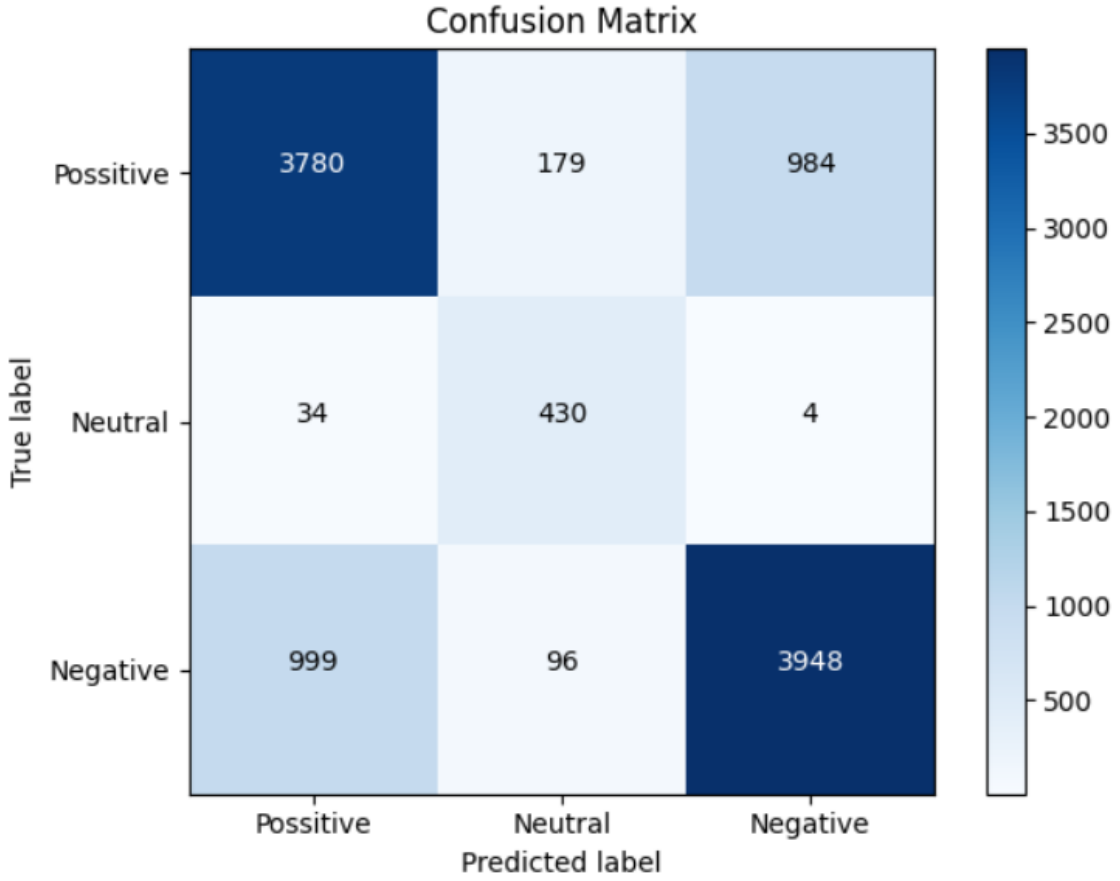
	precision	recall	f1-score	support
Negative	0.73	0.76	0.74	5443
Neutral	0.76	0.39	0.52	497
Positive	0.72	0.72	0.72	5345
accuracy			0.73	11285
macro avg	0.74	0.63	0.66	11285
weighted avg	0.73	0.73	0.72	11285

5.1.2 Naive bayes với Gaussian và word2vec skip-gram

- Số lượng nhãn dự đoán và nhãn thực tế:

```
predict: [4813  705 4936]
reality: [4943  468 5043]
```

- Ma trận nhầm lẫn:



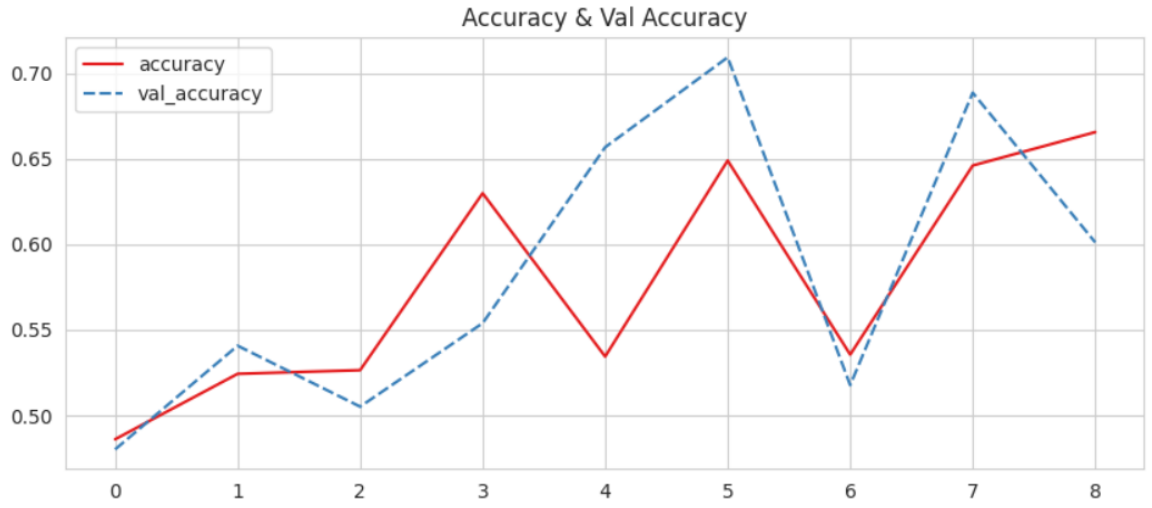
- Giá trị Precision và Recall:

	precision	recall	f1-score	support
negative	0.79	0.76	0.77	4943
neutral	0.61	0.92	0.73	468
positive	0.80	0.78	0.79	5043
accuracy			0.78	10454
macro avg	0.73	0.82	0.77	10454
weighted avg	0.78	0.78	0.78	10454

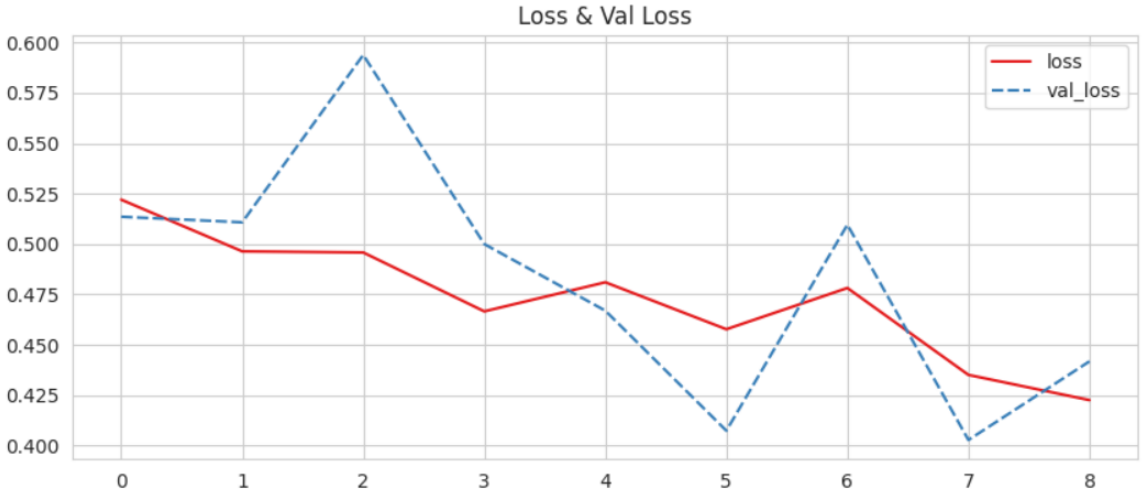
5.2. Kết quả huấn luyện bằng LSTM

5.2.1. LSTM với Word2Vec:

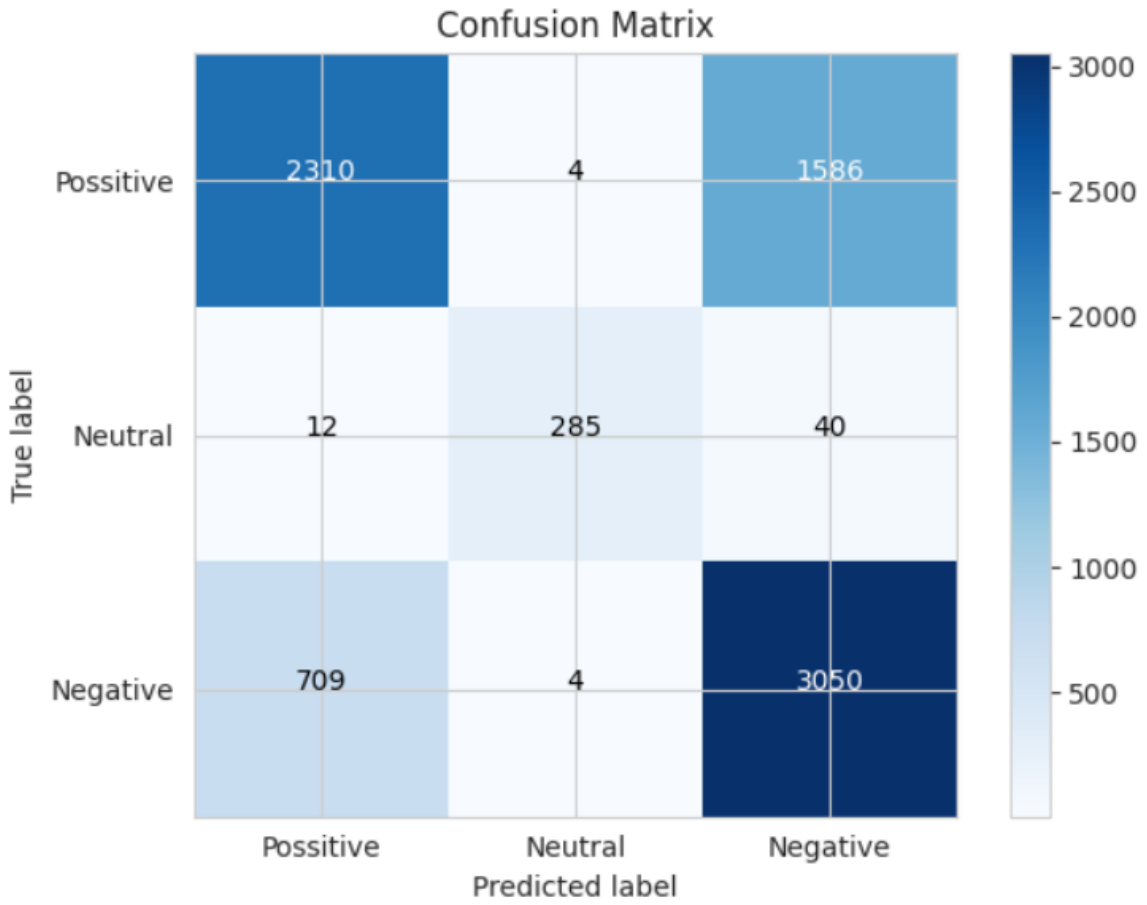
- Biểu đồ các đường accuracy và loss trên mỗi epoch khi huấn luyện:
  + Đường Accuracy:



  + Đường Loss:



- Ma trận nhầm lẫn:

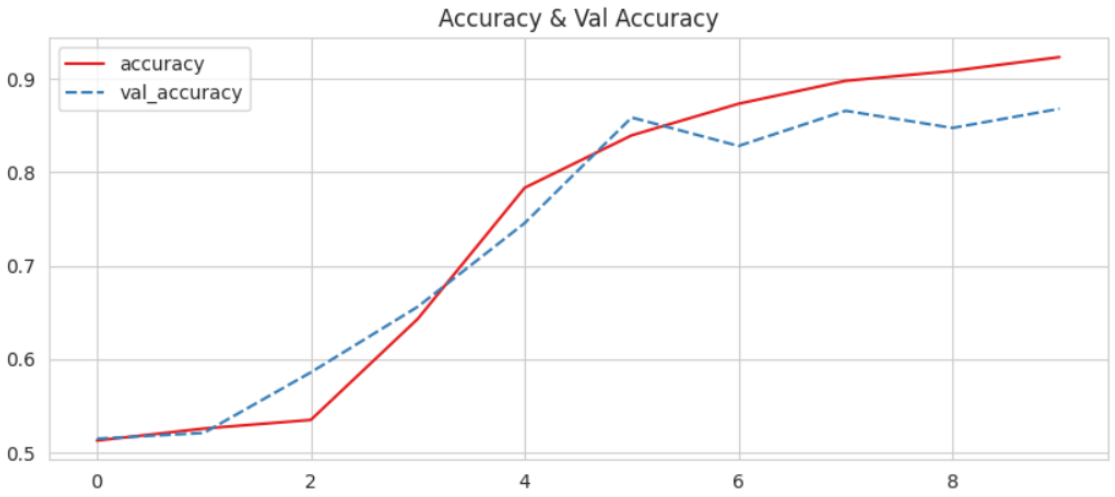


- Giá trị Precision và Recall:

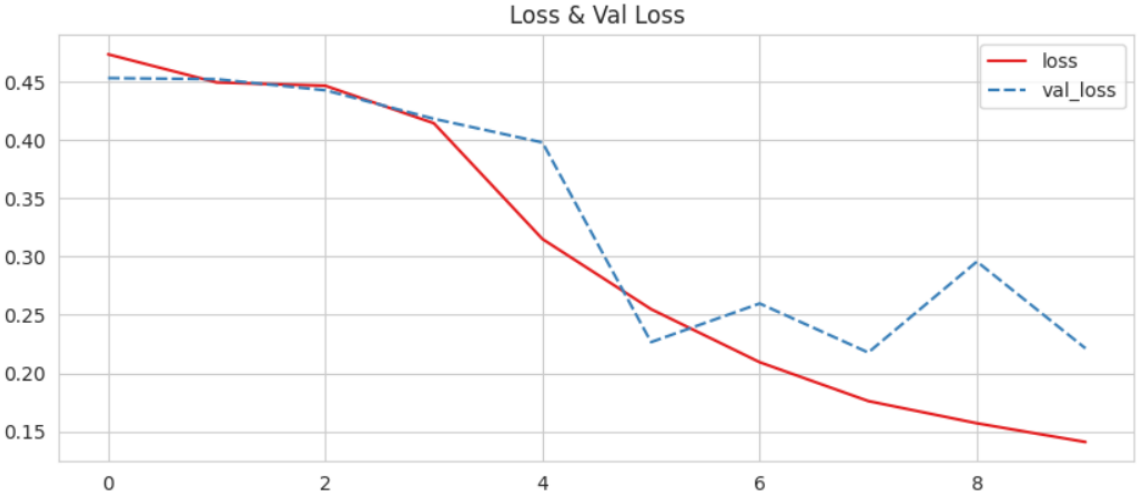
	precision	recall	f1-score	support
negative	0.76	0.59	0.67	3900
neutral	0.97	0.85	0.90	337
positive	0.65	0.81	0.72	3763
accuracy			0.71	8000
macro avg	0.80	0.75	0.76	8000
weighted avg	0.72	0.71	0.70	8000

5.2.2. LSTM với Tokenizer:

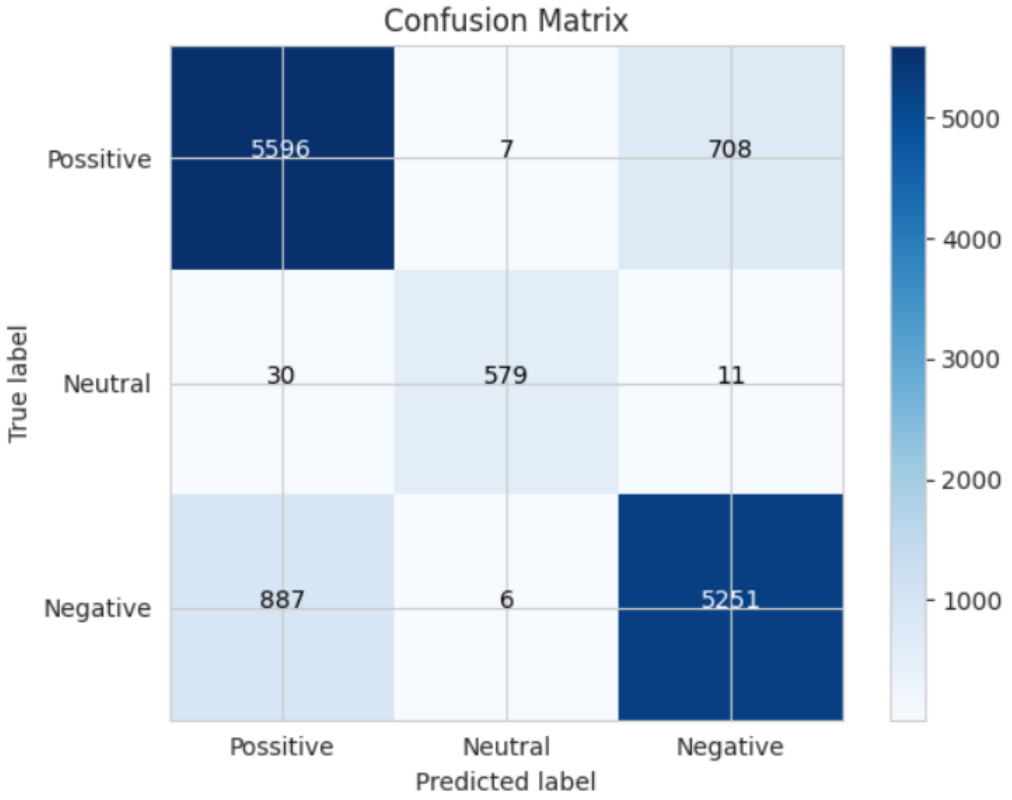
- Biểu đồ các đường accuracy và loss trên mỗi epoch khi huấn luyện:
  + Đường Accuracy:



  + Đường Loss:



- Ma trận nhầm lẫn:



- Giá trị Precision và Recall:

	precision	recall	f1-score	support
negative	0.86	0.89	0.87	6311
neutral	0.98	0.93	0.96	620
positive	0.88	0.85	0.87	6144
accuracy			0.87	13075
macro avg	0.91	0.89	0.90	13075
weighted avg	0.87	0.87	0.87	13075

5.3. So sánh các mô hình

STT	Mô hình nhúng từ	Mô hình huấn luyện	Độ chính xác
1	BOW	Multinomial Naive Bayes	0.73
2	Word2Vec Skip-gram (mean of words)	Gaussian Naive Bayes	0.78
3	Word2Vec Skip-gram	LSTM	0.71
4	Tokenizer + Embedding layer	LSTM	0.87

5.3.1. Nhận xét và so sánh các mô hình

- Phương pháp Naive Bayes với Bag of Words cho kết quả khá tốt, đạt độ chính xác 0.73. Tuy nhiên, độ chính xác của lớp Neutral không cao và F1-score cho lớp Neutral cũng thấp.
- Phương pháp Naive Bayes với Word2Vec skip-gram cho kết quả tốt hơn so với phương pháp Bag of Words, với độ chính xác 0.78. Độ chính xác và F1-score cho lớp Neutral tăng lên đáng kể.
- Phương pháp LSTM với Word2Vec cho kết quả tương đối, với độ chính xác 0.71. Tuy nhiên, độ chính xác cho lớp Negative thấp hơn so với các phương pháp khác.
- Phương pháp LSTM với Tokenizer cho kết quả tốt nhất trong các phương pháp đã được thử, với độ chính xác 0.87. Cả độ chính xác và F1-score đều khá cao cho tất cả các lớp.

- Độ chính xác (accuracy): Các mô hình LSTM (với Tokenizer) và Naive Bayes (với Word2Vec skip-gram) có độ chính xác cao hơn so với mô hình LSTM với Word2Vec và Naive Bayes với Bag of Words.
- Độ phủ (recall): Mô hình Naive Bayes với Word2Vec skip-gram và LSTM với Tokenizer có độ phủ cao hơn so với mô hình Naive Bayes với Bag of Words và LSTM với Word2Vec.
- Độ chính xác của từng lớp (precision): Mô hình LSTM với Tokenizer có độ chính xác cao hơn so với các mô hình khác.
- F1-score: Mô hình LSTM với Tokenizer có F1-score cao hơn so với các mô hình khác.

Tổng quan, phương pháp LSTM với Tokenizer đạt kết quả tốt nhất trong việc phân loại cảm xúc trên đánh giá phim, với độ chính xác 0.87 và F1-score gần như đồng đều cho tất cả các lớp. Phương pháp Naive Bayes với Word2Vec skip-gram cũng cho kết quả tốt, đạt độ chính xác 0.78 và F1-score

đều khá cao. Tuy nhiên, phương pháp Naive Bayes với Bag of Words và LSTM với Word2Vec có độ chính xác và F1-score thấp hơn so với hai phương pháp trên.

5.3.2. Phân tích nguyên nhân

Một số nguyên nhân dẫn đến các kết quả của các mô hình:

- Đặc tính và khả năng của mô hình: Mỗi mô hình có đặc điểm riêng, ví dụ như khả năng xử lý thông tin về thứ tự từ, hiểu ý nghĩa ngữ nghĩa của từ, hoặc xử lý các từ không đồng nhất. Các mô hình có khả năng tốt hơn trong việc mô hình hóa các yếu tố này có thể cho kết quả tốt hơn.
- Kích thước và đa dạng của tập dữ liệu: Kích thước và đa dạng của tập dữ liệu có thể ảnh hưởng đến hiệu suất của mô hình trong quá trình học tập và dự đoán. Nếu tập dữ liệu nhỏ hoặc không đại diện cho đầy đủ các cảm xúc và ngữ cảnh, mô hình có thể gặp khó khăn trong việc học và dự đoán đúng.
- Phương pháp tiền xử lý dữ liệu: Các phương pháp tiền xử lý dữ liệu khác nhau như loại bỏ stop words, stemming/lemmatizing từ, loại bỏ ký tự đặc biệt có thể ảnh hưởng đến chất lượng đặc trưng và dữ liệu đầu vào cho mô hình. Các phương pháp khác nhau có thể ảnh hưởng đến kết quả cuối cùng.
- Thông số và cấu hình mô hình: Các tham số và cấu hình mô hình, chẳng hạn như số lượng epoch, kích thước batch, learning rate, kiến trúc mạng, có thể ảnh hưởng đến hiệu suất của mô hình. Sự lựa chọn không tốt của các tham số và cấu hình có thể dẫn đến kết quả không tốt.
- Sự phân bố không đều của các lớp: Nếu số lượng mẫu trong các lớp khác nhau không đồng đều, mô hình có thể thiên về dự đoán các lớp có số lượng mẫu lớn hơn. Điều này có thể làm giảm hiệu suất của mô hình đối với các lớp có số lượng mẫu ít hơn.

6. Kết luận và hướng phát triển

Trong nghiên cứu này, chúng em đã tiến hành phân tích cảm xúc trên bình luận phim bằng cách áp dụng các mô hình máy học khác nhau như Naive Bayes với Bag of Words, Naive Bayes với Word2Vec skip-gram, LSTM với Word2Vec và LSTM với Tokenizer. Mục tiêu của nghiên cứu là đánh giá và so sánh hiệu suất của các mô hình trong việc phân loại các cảm xúc khác nhau trong bình luận phim.

Dựa trên kết quả đánh giá, chúng em nhận thấy rằng mô hình LSTM với Tokenizer cho kết quả tốt nhất trong số các mô hình được đánh giá. Mô hình này đạt độ chính xác cao và có độ phủ tốt cho tất cả các lớp cảm xúc. Ngoài ra, mô hình Naive Bayes với Word2Vec skip-gram cũng cho kết quả khả quan với độ chính xác và độ phủ cao. Tuy nhiên, mô hình LSTM với Word2Vec và Naive Bayes với Bag of Words có hiệu suất thấp hơn trong một số khía cạnh. Tổng kết lại, nghiên cứu này đã cho thấy mô hình LSTM với Tokenizer và Naive Bayes với Word2Vec skip-gram là hai mô hình có hiệu suất tốt trong phân loại cảm xúc trong bình luận phim.

Nhóm cũng mạnh dạn đề xuất một số hướng phát triển cho đề tài:

Mở rộng tập dữ liệu: Tăng kích thước tập dữ liệu huấn luyện (đặc biệt là dữ liệu nhãn neutral đang còn thiếu sót nhiều) để đảm bảo đại diện cho các cảm xúc và ngữ cảnh phong phú hơn. Điều này có thể đảm bảo rằng mô hình được huấn luyện trên các trường hợp đa dạng và có khả năng tổng quát hóa tốt hơn.

Tối ưu hóa mô hình: Tinh chỉnh các tham số và cấu hình mô hình để đạt hiệu suất tốt hơn. Thử nghiệm với các kiến trúc mô hình khác nhau, điều chỉnh tốc độ học (learning rate), số lượng epoch và kích thước batch để tìm ra cấu hình tốt nhất cho bài toán này.

Khám phá các mô hình học sâu khác: Nghiên cứu và thử nghiệm các mô hình học sâu khác như mạng nơ-ron tái lập (GRU), hoặc mạng nơ-ron biến đổi (Transformer) để xem xét hiệu suất và khả năng tổng quát hóa của chúng trong bài toán phân loại cảm xúc.

Sử dụng kỹ thuật học không giám sát: Nghiên cứu và áp dụng các phương pháp học không giám sát như gom cụm (clustering) và phân loại không giám sát (unsupervised classification) để khám phá cấu trúc và mối quan hệ giữa các cảm xúc trong dữ liệu mà không cần nhãn dữ liệu.

TÀI LIỆU THAM KHẢO

[1]. Slide Khai phá Web kỳ 20222, thầy Nguyễn Kiêm Hiếu, SoICT, HUST.
[2]. Trang web hướng dẫn Tiền xử lý dữ liệu: How to Prepare Movie Review Data for Sentiment Analysis (Text Classification).
[3]. Trang tài liệu hướng dẫn nhúng dữ liệu: NLP Tutorial | Word2Vec & Glove Embeddings
[4]. Trang tài liệu về Naive Bayes trong bài toán phân loại: Naive Bayes Classifier
[5]. Trang tài liệu về LSTM cho bài toán Sentiment Analysis: Sentiment Analysis with LSTM