

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



NHÓM 4
BÁO CÁO
BTL Xử Lý Ngôn Ngữ Tự Nhiên

ĐỀ TÀI: PHÂN TÍCH CẢM XÚC

Sinh viên thực hiện: Mai Phạm Lan Anh - 0263367
Lê Việt Hùng - 0174367
Văn Đức Anh - 0279567
Nguyễn Hoài Nam - 0055167

Giảng viên hướng dẫn: Thầy Nguyễn Đình Quý

Hà Nội, 5-2025

LỜI CẢM ƠN

Lời đầu tiên, nhóm 4 lớp Xử lý ngôn ngữ tự nhiên 67CS1 xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Đình Quý – người đã tận tình hướng dẫn, chỉ bảo nhóm trong suốt quá trình thực hiện đồ án. Thầy không chỉ mang đến cho chúng em những kiến thức quý báu về môn học mà còn chia sẻ những kinh nghiệm thực tế giúp chúng em hoàn thiện kỹ năng và kiến thức để áp dụng vào đồ án.

Chúng em xin cam đoan, tất cả nội dung của bài tập lớn đều được tất cả thành viên nhóm 4 lên ý tưởng học hỏi, tham khảo và hoàn thiện.

Chúng em xin chân thành cảm ơn!

Mục lục

I	Giới thiệu	3
1	Giới thiệu đề tài	3
2	Cơ sở lý thuyết	3
2.1	Xử lý ngôn ngữ tự nhiên (NLP)	3
2.2	Word Embedding	3
2.3	Mạng LSTM (Long Short-Term Memory)	3
2.4	BiLSTM (Bidirectional LSTM)	4
2.5	Hàm mất mát	4
II	Hướng thực hiện đề tài	4
1	Tìm hiểu và phân tích dữ liệu	4
2	Tiền xử lý văn bản	4
3	Chia dữ liệu huấn luyện và kiểm thử	5
4	Xây dựng và huấn luyện mô hình	5
5	Đánh giá mô hình	5
III	Mục tiêu đề ra	6
IV	Nội dung thực hiện	6
1	Dữ liệu	6
2	Mô hình sử dụng	7
V	Quy trình thực hiện	8
1	Thu thập và khảo sát dữ liệu	8
2	Các siêu tham số	9
3	Huấn luyện mô hình	9
4	Đánh giá mô hình	15
5	Ma trận nhầm lẫn	17
6	Biểu đồ Loss và Accuracy	18
7	Dự đoán cảm xúc cho 1 câu	19
VI	Hướng cải thiện	20
VII	Kết luận	20

I Giới thiệu

1 Giới thiệu đề tài

Phân tích cảm xúc (Sentiment Analysis) là một lĩnh vực thuộc xử lý ngôn ngữ tự nhiên (NLP), với mục tiêu xác định thái độ, ý kiến hoặc cảm xúc của người viết thông qua nội dung văn bản. Nó giúp máy tính hiểu được cảm xúc ẩn sau câu chữ – liệu một câu văn là tích cực, trung tính hay tiêu cực.

Ứng dụng thực tế:

- Đánh giá mức độ hài lòng của khách hàng đối với sản phẩm/ dịch vụ
- Phân tích bình luận mạng xã hội để nắm bắt xu hướng dư luận.
- Hệ thống gợi ý, chăm sóc khách hàng tự động, chatbot thông minh,...

2 Cơ sở lý thuyết

2.1 Xử lý ngôn ngữ tự nhiên (NLP)

Xử lý ngôn ngữ tự nhiên là lĩnh vực nghiên cứu giúp máy tính hiểu, diễn giải và tạo ra ngôn ngữ của con người. Trong đề tài này, các kỹ thuật NLP được dùng để tiền xử lý văn bản và chuyển đổi thành dạng mà máy tính có thể xử lý.

2.2 Word Embedding

Đây là phương pháp chuyển đổi từ ngữ thành các vector số có nghĩa. Mỗi từ sẽ được biểu diễn bằng một vector nhiều chiều (thường là 100-300 chiều). Các từ có ngữ nghĩa tương đồng sẽ có vector gần nhau trong không gian vector. Ví dụ: GloVe, Word2Vec,...

2.3 Mạng LSTM (Long Short-Term Memory)

LSTM là một kiến trúc mạng RNN cải tiến, có khả năng ghi nhớ và xử lý thông tin lâu dài tốt hơn RNN truyền thống. LSTM giải quyết vấn đề vanishing gradient, rất hiệu quả với dữ liệu chuỗi như ngôn ngữ.

2.4 BiLSTM (Bidirectional LSTM)

BiLSTM là phiên bản mở rộng của LSTM, trong đó mô hình được huấn luyện theo hai chiều:

- Forward: từ trái sang phải.
- Backward: từ phải sang trái.

Điều này giúp mô hình hiểu ngữ cảnh tốt hơn vì nó xét được thông tin cả trước và sau của một từ trong câu.

2.5 Hàm mất mát

Hàm mất mát được sử dụng là `sparse_categorical_crossentropy`, phù hợp với bài toán phân loại nhiều lớp (multi-class classification) mà nhãn là số nguyên (0, 1, 2).

II Hướng thực hiện đề tài

1 Tìm hiểu và phân tích dữ liệu

- Lựa chọn và phân tích tập dữ liệu Amazon Fine Food Reviews - một tập dữ liệu lớn chứa các đánh giá văn bản thực tế từ người dùng.
- Rút trích các trường thông tin cần thiết, bao gồm Text (nội dung đánh giá) và Score (điểm đánh giá), sau đó gán nhãn cảm xúc theo ba mức độ
 - Từ 1 đến 2: tiêu cực (negative) - label = 0
 - 3: trung lập (neutral) - label = 1
 - Từ 4 đến 5: tích cực (positive) - label = 2

2 Tiền xử lý văn bản

- Làm sạch dữ liệu bằng cách:
 - Loại bỏ thẻ HTML, ký tự đặc biệt, số và chuyển toàn bộ văn bản về chữ thường

- Loại bỏ các từ dừng (stopwords) bằng thư viện nltk
- Ánh xạ văn bản thành dạng số bằng Tokenizer và pad_sequences để đưa vào mô hình học sâu.

3 Chia dữ liệu huấn luyện và kiểm thử

- Chia dữ liệu thành 3 tập:
 - Tập huấn luyện (training set): 64% dữ liệu
 - Tập kiểm tra (validation set): 16% dữ liệu
 - Tập kiểm thử (test set): 20% dữ liệu
- Đảm bảo phân phối nhãn đều giữa các tập bằng phương pháp phân tầng (stratify)

4 Xây dựng và huấn luyện mô hình

- Thiết kế mô hình học sâu sử dụng kiến trúc Bi-LSTM (Bidirectional LSTM) gồm:
 - Lớp Embedding tự huấn luyện
 - Hai lớp Bi-LSTM để nắm bắt ngữ cảnh hai chiều
 - Các lớp Dropout, Dense để giảm overfitting và tăng khả năng khái quát hóa
- Sử dụng class_weight để cân bằng dữ liệu giữa các lớp
- Huấn luyện mô hình với early stopping để tránh học quá mức (overfitting)

5 Đánh giá mô hình

- Dự đoán trên tập dữ liệu kiểm thử
- Đánh giá hiệu năng bằng:
 - Accuracy và loss trên tập val/test
 - Classification report (Precision, Recall, F1-score)
 - Confusion matrix trực quan hóa nhầm lẫn giữa các lớp

III Mục tiêu đề ra

Đề tài hướng đến việc xây dựng một hệ thống phân tích cảm xúc có khả năng tự động phân loại các đánh giá sản phẩm trên nền tảng thương mại điện tử thành ba nhóm cảm xúc: tiêu cực, trung lập và tích cực. Mục tiêu cụ thể bao gồm:

- Tiền xử lý và làm sạch dữ liệu đánh giá văn bản để loại bỏ nhiễu và chuẩn hóa đầu vào.
- Gán nhãn cảm xúc phù hợp với từng mức độ đánh giá (score).
- Biến đổi văn bản thành các chuỗi số thông qua GloVe word embedding để mô hình có thể hiểu và xử lý dữ liệu ngôn ngữ tự nhiên.
- Xây dựng và huấn luyện mô hình Bi-LSTM để khai thác đặc trưng ngữ cảnh từ chuỗi văn bản.
- Đánh giá hiệu quả mô hình thông qua các chỉ số như accuracy, confusion matrix, và classification report.

IV Nội dung thực hiện

1 Dữ liệu

Dữ liệu trong bài tập lớn này là một tập dữ liệu về các đánh giá sản phẩm của người tiêu dùng trên sàn thương mại điện tử Amazon. Và dữ liệu được lấy từ các nguồn lấy dữ liệu lớn như Kaggle. Bộ dữ liệu được lựa chọn: Link: <https://www.kaggle.com/code/robikscube/sentiment-analysis-py>
select=Reviews.csv

id	Productid	Userid	ProfileId	Helpful	Helpful Score	Time	Summary	Text
1	B001E4KFA3	G3GXH7	delmart	1	1	5	1E+09	Good Qual I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a proc
2	B00813GFA1	D87F6Z	dill pa	0	0	1	1E+09	Not as Adv Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to i
3	B000LQOIA	ABXLMWJ	Natalia	1	1	4	1E+09	"Delight" is This is a confection that has been around a few centuries. It is a light, pillowy citrus gelatin with nuts - in this case Filberts. And it is cut into tiny squares and
4	B000UAOC	A395BOR	Karl	3	3	2	1E+09	Cough Mel If you are looking for the secret ingredient in Robitussin I believe I have found it. I got this in addition to the Root Beer Extract I ordered (which was good) an
5	B006K2ZZ	A1UQRSC	Michael	0	0	5	1E+09	Great Taffy Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery was very quick. If your a taffy lover, this is a deal.
6	B006K2ZZ	ADT0SRK	Twoapt	0	0	4	1E+09	Nice Taffy I got a wild hair for taffy and ordered this five pound bag. The taffy was all very enjoyable with many flavors: watermelon, root beer, melon, peppermint, grap
7	B006K2ZZ	A1SP2KVp	David C	0	0	5	1E+09	Great! Jus This saltwater taffy had great flavors and was very soft and chewy. Each candy was individually wrapped well. None of the candies were stuck together, w
8	B006K2ZZ	A3JRG0V	Pamela	0	0	5	1E+09	Wonderful This taffy is so good. It is very soft and chewy. The flavors are amazing. I would definitely recommend you buying it. Very satisfying!!
9	B000E7L2	L1MZVQ9	R. Jam	1	1	5	1E+09	Yay Barley Right now I'm mostly just sprouting this so my cats can eat the grass. They love it. I rotate it around with Wheatgrass and Rye too
10	B00171AP	A21BT40V	Carol A	0	0	5	1E+09	Healthy Dk This is a very healthy dog food. Good for their digestion. Also good for small puppies. My dog eats her required amount at every feeding.
11	B0001PB9	A3HDKO7	Canadi	1	1	5	1E+09	The Best I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavour of this hot sauce makes it one of a kind! We picked u
12	B0009XLV	A2725IB4	A Poem	4	4	5	1E+09	My cats L One of my boys needed to lose some weight and the other didn't. I put this food on the floor for the chubby guy, and the protein-rich, no by-product food up
13	B0009XLV	A3Z7PCT2	L T	1	1	1	1E+09	My Cats A My cats have been happily eating Felidae Platinum for more than two years. I just got a new bag and the shape of the food is different. They tried the new fo
14	B001GVIS	A18ECVX2	willie "tr	2	2	4	1E+09	fresh and good flavor! these came securely packed... they were fresh and delicious! I love these Twizzlers!
15	B001GVIS	A2MUGFV	Lyndie "	4	5	5	1E+09	Strawberry The Strawberry Twizzlers are my guilty pleasure - yummy. Six pounds will be around for a while with my son and I.
16	B001GVIS	A1CZK3Cf	Brian A	4	5	5	1E+09	Lots of twi My daughter loves twizzlers and this shipment of six pounds really hit the spot. It's exactly what you would expect...six packages of strawberry twizzlers.
17	B001GVIS	A3KLVF6	Erica N	0	0	2	1E+09	poor taste I love eating them and they are good for watching TV and looking at movies! It is not too sweet. I like to transfer them to a zip lock baggie so they stay fresh :
18	B001GVIS	AFKW14U	Becca	0	0	5	1E+09	Love it! I am very satisfied with my Twizzler purchase. I shared these with others and we have all enjoyed them. I will definitely be ordering more.
19	B001GVIS	A2A9X58G	Wolfee	0	0	5	1E+09	GREAT S Twizzlers, Strawberry my childhood favorite candy, made in Lancaster Pennsylvania by Y & S Candies, Inc. one of the oldest confectionery Firms in the Unit
20	B001GVIS	A3iV7CL2	Greg	0	0	5	1E+09	Home deli Candy was delivered very fast and was purchased at a reasonable price. I was home bound and unable to get to a store so this was perfect for me.
21	B001GVIS	A1V0OKG	Gmon2e	0	0	5	1E+09	Always fre My husband is a Twizzlers addict. We've bought these many times from Amazon because we're government employees living overseas and can't get them il
22	B001GVIS	AZOF9E17	Tammy	0	0	5	1E+09	TWIZZLEF I bought these for my husband who is currently overseas. He loves these, and apparently his staff likes them also. There are generous amounts of Twi
23	B001GVIS	ARV9QL4	Charles	0	0	5	1E+09	Delicious f I can remember buying this candy as a kid and the quality hasn't dropped in all these years. Still a superb product you won't be disappointed with.
24	B001GVIS	AJ613OLZ	Mare's	0	0	5	1E+09	Twizzlers I love this candy. After weight watchers I had to cut back but still have a craving for it.
25	B001GVIS	A22P2J09	S. Cab	0	0	5	1E+09	Please se I have lived out of the US for over 7 yrs now, and I so miss my Twizzlers!! When I go back to visit or someone visits me, I always stock up. All I can say is Y
26	B001GVIS	A3FONPR	Debora	0	0	5	1E+09	Twizzlers - Product received is as advertised. Twizzlers, Strawberry, 16-Ounce Bags (Pack of 6
27	B001GVIS	A3RXAU2	lady21	0	1	1	1E+09	Nasty No f The candy is just red, No flavor. Just plan and chewy. I would never buy them again
28	B001GVIS	AAAS38B	Heathe	0	1	4	1E+09	Great Bar I was so glad Amazon carried these batteries. I have a hard time finding them elsewhere because they are such a unique size. I need them for my garage d
29	B00144C1	A2F4LZV	DaisyH	0	0	5	1E+09	YUMMY! I got this for my Mum who is not diabetic but needs to watch her sugar intake, and my father who simply chooses to limit unnecessary sugar intake - she's th

- Dữ liệu gồm hơn 500k dòng đánh giá của khách hàng, mỗi dòng chứa nhiều thông tin như: ID sản phẩm, tên người đánh giá, điểm đánh giá, nội dung đánh giá, thời gian,...
- Dữ liệu sử dụng cho đề tài được rút trích các trường thông tin cần thiết "Score" và "Text" và gán nhãn theo 3 mức độ: negative, neutral, positive.
- Chia dữ liệu thành 3 tập:
 - Tập huấn luyện (training set): 64% dữ liệu
 - Tập kiểm tra (validation set): 16% dữ liệu
 - Tập kiểm thử (test set): 20% dữ liệu

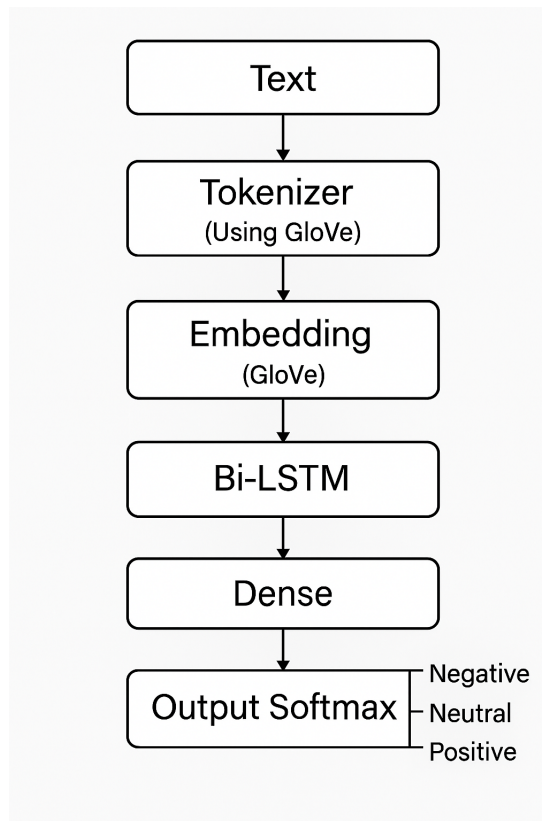
2 Mô hình sử dụng

Đối với đề tài phân tích cảm xúc, việc sử dụng mô hình LSTM là một lựa chọn ưu tiên. Trong đề tài này, nhóm sử dụng mô hình Bi-directional Long Short-Term Memory (Bi-LSTM) kết hợp với lớp embedding được huấn luyện trong quá trình xây dựng để giải quyết bài toán phân tích cảm xúc văn bản đánh giá sản phẩm.

Mô hình bao gồm các thành phần chính như sau:

- Lớp embedding tự huấn luyện: Tầng này chuyển đổi các từ thành vector số, với kích thước từ vựng là 20,000, chiều embedding là 100 và độ dài tối đa của câu là 200. Tầng Embedding được thiết lập để tự học các vector embedding từ dữ liệu huấn luyện, thay vì sử dụng embedding được huấn luyện sẵn như GloVe.

- LSTM hai chiều (Bi-LSTM): Là phần cốt lõi trong mô hình, có khả năng ghi nhớ và học được ngữ cảnh của văn bản theo cả chiều từ trái sang phải và ngược lại. Điều này đặc biệt quan trọng trong ngôn ngữ tự nhiên, vì ý nghĩa của một từ có thể phụ thuộc vào các từ ở cả trước và sau nó.
- Cả lớp Dense và Dropout: Sau khi LSTM, biểu diễn ngữ cảnh được đưa qua các lớp fully connected (Dense) với các hàm kích hoạt ReLU để học các đặc trưng phân loại. Dropout được sử dụng xen kẽ nhằm giảm hiện tượng overfitting.
- Lớp đầu ra (Output Layer): Sử dụng hàm kích hoạt Softmax để phân loại đầu ra thành ba lớp cảm xúc: tiêu cực (negative), trung lập (neutral) và tích cực (positive)



V Quy trình thực hiện

1 Thu thập và khảo sát dữ liệu

Chia tập dữ liệu: tập huấn luyện, tập kiểm thử, tập kiểm tra.

- Ngẫu nhiên hóa: Chia dữ liệu một cách ngẫu nhiên để đảm bảo mỗi tập đại diện đầy đủ cho toàn bộ dữ liệu.
- Không trùng lặp: Đảm bảo rằng các mẫu trong tập test không bị trùng lặp với tập train và val.

2 Các siêu tham số

Bảng 1: Các siêu tham số huấn luyện của mô hình

Tên siêu tham số	Giá trị	Mô tả
Số lượng từ vựng tối đa (vocab_size)	20.000	Giới hạn số lượng từ phổ biến nhất được sử dụng trong quá trình huấn luyện.
Độ dài tối đa mỗi câu (maxlen)	100 từ	Các câu được cắt hoặc thêm padding để có độ dài cố định.
Kích thước embedding (embedding_dim)	128	Số chiều của vector biểu diễn từ.
Số đơn vị LSTM (units)	64	Số lượng đơn vị ẩn trong mỗi lớp LSTM hai chiều.
Số lớp BiLSTM	2	Hai lớp BiLSTM giúp trích xuất tốt hơn ngữ cảnh theo cả hai hướng.
Tỷ lệ dropout	0.3	Tỷ lệ dropout áp dụng sau mỗi lớp để giảm hiện tượng overfitting.
Bộ tối ưu hóa (optimizer)	Adam	Tối ưu hóa dựa trên gradient descent với khả năng thích ứng.
Tốc độ học (learning_rate)	0.001	Tốc độ cập nhật trọng số trong quá trình học.
Hàm mất mát	sparse_categorical_crossentropy	Phù hợp với bài toán phân loại nhiều lớp với nhãn dạng số nguyên.
Số vòng lặp huấn luyện (epochs)	5-10	Số lần duyệt toàn bộ dữ liệu huấn luyện.
Kích thước batch (batch_size)	128	Số mẫu xử lý trong mỗi lần cập nhật trọng số.
Tỷ lệ kiểm tra (validation_split)	0.2	Tỷ lệ dữ liệu được sử dụng để kiểm tra trong quá trình huấn luyện.

3 Huấn luyện mô hình

Thực hiện huấn luyện mô hình trên Google Colab

- Khai báo các thư viện cần thiết

```

import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from sklearn.model_selection import
    train_test_split
from sklearn.metrics import classification_report
    , confusion_matrix
from sklearn.utils import class_weight
from tensorflow.keras.preprocessing.text import
    Tokenizer
from tensorflow.keras.preprocessing.sequence
    import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding,
    Bidirectional, LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import
    EarlyStopping
import seaborn as sns
import matplotlib.pyplot as plt
import pickle

```

- Đọc dữ liệu và tiền xử lý dữ liệu

Tiền xử lý dữ liệu, làm sạch văn bản

```

✓ [54] def clean_text(text):
13s      text = text.lower()
      text = re.sub(r'<[>]+>', ' ', text)
      text = re.sub(r'[^a-z\s]', ' ', text) #xóa ký tự không phải chữ cái
      text = ' '.join(word for word in text.split() if word not in stop_words)
      return text

      df['Text'] = df['Text'].apply(clean_text)

```

Chuyển đổi điểm thành nhãn

```

✓ [55] def convert_score(score):
0s      if score <= 2:
          return 0 # negative
      elif score == 3:
          return 1 # neutral
      else:
          return 2 # positive

      df['label'] = df['Score'].apply(convert_score)

```

Ta chỉ lấy dữ liệu ở cột Text và Score, sau đó dữ liệu được làm sạch: chuyển thành chữ thường, loại bỏ HTML, dấu câu, ký tự đặc biệt và

stopwords

Sau đó dữ liệu được gán nhãn cảm xúc: biến đổi điểm số thành 3 nhãn:
0-tiêu cực(1-2), 1-trung tính(3), 3-tích cực(4-5)

- Chia dữ liệu thành train, val, test

Chia dữ liệu: 80% train_val, 20% test. Từ train_val: 80% train, 20% validation

```
[56] train_val_df, test_df = train_test_split(df, test_size=0.2, stratify=df['label'], random_state=42)
      train_df, val_df = train_test_split(train_val_df, test_size=0.2, stratify=train_val_df['label'], random_state=42)
```

Chuẩn bị dữ liệu

```
[57] X_train = train_df['Text'].values
      y_train = train_df['label'].values
      X_val = val_df['Text'].values
      y_val = val_df['label'].values
      X_test = test_df['Text'].values
      y_test = test_df['label'].values
```

stratify: đảm bảo tỉ lệ nhãn đồng đều giữa các tập.

- Thiết lập tham số

Thiết lập tham số

```
[62] vocab_size = 20000
      max_len = 200
      embedding_dim = 100
```

- vocab_size: giới hạn số từ tối đa trong từ điển là 20000 từ(chỉ giữ từ phổ biến nhất).
- max_len: độ dài tối đa của mỗi câu là 200 từ (câu dài hơn sẽ bị cắt, câu ngắn hơn sẽ được đệm)
- embedding_dim: kích thước vector embedding cho mỗi từ là 100

- Tokenizer

Tokenization và padding

```
[63] tokenizer = Tokenizer(num_words=vocab_size, oov_token="<OOV>")
      tokenizer.fit_on_texts(X_train)

      X_train_seq = tokenizer.texts_to_sequences(X_train)
      X_val_seq = tokenizer.texts_to_sequences(X_val)
      X_test_seq = tokenizer.texts_to_sequences(X_test)

      X_train_pad = pad_sequences(X_train_seq, maxlen=max_len, padding='post')
      X_val_pad = pad_sequences(X_val_seq, maxlen=max_len, padding='post')
      X_test_pad = pad_sequences(X_test_seq, maxlen=max_len, padding='post')
```

oov_token = vocab_size: từ không có trong từ điển sẽ được thay bằng token <OOV>

Xây dựng từ điển từ tập train, ánh xạ mỗi từ thành một số nguyên. Ví dụ: "good" -> 10, "product" -> 15

Sau đó `texts_to_sequences` sẽ chuyển mỗi câu thành chuỗi số. VD: [10, 15]

Đảm bảo rằng tất cả chuỗi có độ dài bằng `max_len`, nếu không bằng thì `padding='post'` sẽ đệm số 0 vào cuối chuỗi. VD: [10, 15, 0,...,0].

Và nếu chuỗi dài hơn 200 sẽ bị cắt bớt.

- Tính Class Weights để xử lý mất cân bằng

```
Tính class weight để xử lý mất cân bằng lớp
[64] class_weights = class_weight.compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)
      class_weights_dict = dict(enumerate(class_weights))
```

`compute_class_weight`:

- `balanced`: tính trọng số theo công thức:

$$\text{weight}_i = \frac{\text{Tổng số mẫu}}{\text{Số lớp} \times \text{Số mẫu của lớp } i}$$

- `classes=np.unique(y_train)`: các lớp(0, 1, 2)
- `y=y_train`: nhãn của tập train

VD: train có 64000 mẫu: 40000 positive, 16 neutral, 8000 negative.
Thì trọng số

$$\text{Positive: } \frac{64,000}{3 \times 40,000} \approx 0.53$$

$$\text{Neutral: } \frac{64,000}{3 \times 16,000} \approx 1.33$$


$$\text{Negative: } \frac{64,000}{3 \times 8,000} \approx 2.67$$

`class_weights_dict`: chuyển thành từ điển: 0: 2.67, 1: 1.33, 2: 0.5353

- Xây dựng mô hình Bi-LSTM

Xây dựng mô hình

```
[65] model = Sequential([
    Embedding(vocab_size, embedding_dim, input_length=max_len, trainable=True), # Embedding tự huấn luyện
    Bidirectional(LSTM(128, return_sequences=True)),
    Bidirectional(LSTM(64)),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax') # 3 lớp: negative, neutral, positive
])
```

 /usr/local/lib/python3.11/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it. warnings.warn()

- Sequential: Xây dựng mô hình tuần tự, các tầng được xếp chồng lên nhau.
- Tầng embedding tự học từ dữ liệu
- Tầng Bidirectional(LSTM(128, return_sequences=True)):
 - LSTM(128): Tầng LSTM với 128 đơn vị (hidden state có kích thước 128).
 - return_sequences=True: Trả về đầu ra cho mỗi từ trong chuỗi (200 từ).
 - Bidirectional: Kết hợp hai LSTM (forward và backward), nên đầu ra có kích thước 256 (128 * 2).
 - Đầu ra: Ma trận (200, 256).
- Tầng Bidirectional(LSTM(64))
 - LSTM(64): Tầng LSTM với 64 đơn vị.
 - return_sequences=False: Chỉ trả về đầu ra cuối cùng (tổng hợp toàn bộ chuỗi).
 - Đầu ra: Vector (128,) (64 * 2).

- Biên dịch mô hình

Biên dịch mô hình

```
[84] model.compile(loss='sparse_categorical_crossentropy', optimizer=Adam(learning_rate=1e-3), metrics=['accuracy'])  
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 200, 100)	2,000,000
bidirectional_2 (Bidirectional)	(None, 200, 256)	234,496
bidirectional_3 (Bidirectional)	(None, 128)	164,352
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16,512
dropout_4 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8,256
dropout_5 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 3)	195

Total params: 2,423,811 (9.25 MB)
Trainable params: 2,423,811 (9.25 MB)
Non-trainable params: 0 (0.00 B)

`loss='sparse_categorical_crossentropy'`: Hàm mất mát cho bài toán phân loại đa lớp, phù hợp khi nhãn là số nguyên (0, 1, 2).
`optimizer=Adam(learning_rate=1e-3)`: Sử dụng Adam với learning rate 0.001.
`metrics=['accuracy']`: Theo dõi độ chính xác trong quá trình huấn luyện.
`model.summary()`: In cấu trúc mô hình, bao gồm số tham số của từng tầng.

- Thiết lập early

Thiết lập early stopping

```
[67] early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
```

Mô hình sẽ dừng lại quá trình huấn luyện sớm nếu không còn cải thiện hiệu suất trên tập val sau 1 số epoch nhất định

- Huấn luyện mô hình

Huấn luyện mô hình

```
[68] history = model.fit(X_train_pad, y_train,
                        validation_data=(X_val_pad, y_val),
                        epochs=10,
                        batch_size=128,
                        callbacks=[early_stopping],
                        class_weight=class_weights_dict)
```

```
Epoch 1/10
2843/2843 ————— 180s 62ms/step - accuracy: 0.6955 - loss: 0.8228 - val_accuracy: 0.7551 - val_loss: 0.6063
Epoch 2/10
2843/2843 ————— 207s 64ms/step - accuracy: 0.8050 - loss: 0.5829 - val_accuracy: 0.8131 - val_loss: 0.4852
Epoch 3/10
2843/2843 ————— 196s 61ms/step - accuracy: 0.8403 - loss: 0.4761 - val_accuracy: 0.8213 - val_loss: 0.4664
Epoch 4/10
2843/2843 ————— 181s 64ms/step - accuracy: 0.8651 - loss: 0.3932 - val_accuracy: 0.8304 - val_loss: 0.4487
Epoch 5/10
2843/2843 ————— 196s 62ms/step - accuracy: 0.8879 - loss: 0.3221 - val_accuracy: 0.8403 - val_loss: 0.4909
Epoch 6/10
2843/2843 ————— 181s 63ms/step - accuracy: 0.9061 - loss: 0.2634 - val_accuracy: 0.8310 - val_loss: 0.4674
Epoch 7/10
2843/2843 ————— 202s 64ms/step - accuracy: 0.9174 - loss: 0.2210 - val_accuracy: 0.8583 - val_loss: 0.4873
```

Dùng history để lưu trữ lịch sử huấn luyện (loss, accuracy, val_loss, val_accuracy) để vẽ biểu đồ.

4 Đánh giá mô hình

Dự đoán xác suất cho từng lớp trên tập test

Dự đoán trên tập test

```
[69] y_pred = model.predict(X_test_pad)
     y_pred_classes = y_pred.argmax(axis=1)
```

```
3553/3553 ————— 44s 12ms/step
```

Báo cáo phân loại

```
[70] print("Classification Report:")
     print(classification_report(y_test, y_pred_classes, target_names=['negative', 'neutral', 'positive']))
```

```
Classification Report:
              precision    recall  f1-score   support

 negative       0.73       0.78       0.75      16407
  neutral       0.33       0.69       0.44       8528
   positive       0.97       0.85       0.91      88756

 accuracy              0.68
 macro avg              0.68       0.77       0.70      113691
 weighted avg          0.89       0.83       0.85      113691
```

Hiển thị P, R, F1-Score cho từng lớp giúp đánh giá hiệu suất chi tiết.

- negative
 - precision = 0.73 → Trong các mẫu mô hình dự đoán là “negative”, có 73% là đúng.
 - recall = 0.78 → Trong tất cả các mẫu thực sự là “negative”, mô hình bắt đúng được 78%.
 - f1-score = 0.75 → Hiệu suất tổng hợp tốt.

- support = 16,407 → Có 16,407 mẫu “negative” trong tập test.
- neutral
 - precision = 0.33 → Chỉ 33% trong số các mẫu dự đoán là “neutral” là đúng.
 - recall = 0.69 → Mô hình nhận diện được 69% trong số thực sự là “neutral”.
 - f1-score = 0.44 → Hiệu suất tổng thể khá kém.
 - support = 8,528 → Lớp này có ít dữ liệu và dễ bị nhầm lẫn.
- positive
 - precision = 0.97 → Dự đoán cực kỳ chính xác cho “positive”.
 - recall = 0.85 → Nhận diện được 85% mẫu “positive”.
 - f1-score = 0.91 → Mô hình hoạt động rất tốt ở lớp này.
 - support = 88,756 → Chiếm phần lớn dữ liệu → lớp mất cân bằng!

Tổng thể:

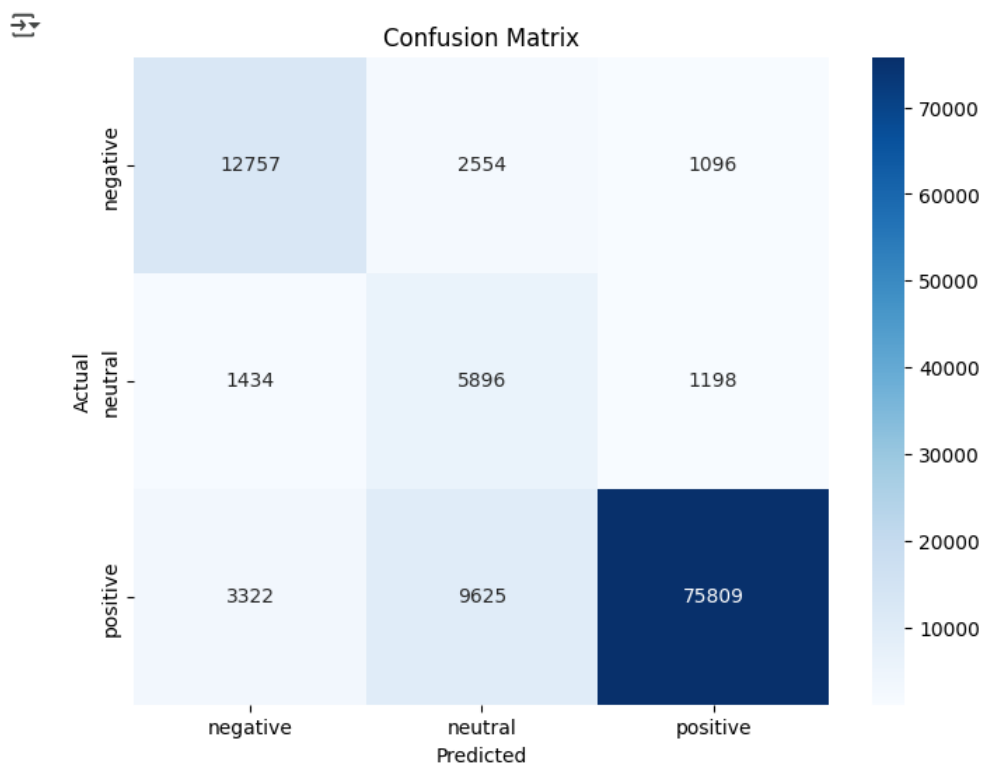
- accuracy = 0.83 → Mô hình đúng 83% trên toàn bộ tập test.
- macro avg:
 - Trung bình không trọng số của các lớp.
 - macro F1 = 0.70 → đánh giá công bằng giữa các lớp.
- weighted avg:
 - Trung bình có trọng số theo số mẫu mỗi lớp.
 - F1 = 0.85 → khá cao vì lớp positive chiếm ưu thế.

Nhận xét:

- Mô hình hoạt động rất tốt với lớp positive, nhưng yếu với lớp neutral.
- Lớp neutral có precision thấp → mô hình dễ nhầm lớp khác thành neutral.

- Đây là dấu hiệu mất cân bằng dữ liệu – mô hình thiên lệch về lớp có nhiều mẫu (positive).
- Cần cải thiện lớp neutral bằng cách:
 - Tăng dữ liệu trung lập.
 - Điều chỉnh trọng số lớp

5 Ma trận nhầm lẫn



Cột là dự đoán, hàng là thực tế.

Đường chéo chính là số lượng dự đoán đúng:

- negative → negative: 12,757
- neutral → neutral: 5,896
- positive → positive: 75,809

Các ô ngoài đường chéo là dự đoán sai (nhầm lẫn):

- 2,554 mẫu negative bị nhầm thành neutral.

- 1,434 mẫu neutral bị nhầm thành negative.
- 9,625 mẫu positive bị nhầm thành neutral → đây là nhầm lẫn lớn nhất.

Nhận xét: Mô hình nhận diện lớp positive rất tốt với 75,809 đúng -> Recall rất cao nhưng cũng bị nhầm nhiều thành neutral

Mô hình nhận diện lớp negative cũng tốt và cũng bị nhầm chủ yếu sang neutral

Mô hình khó nhận diện lớp neutral nhất: vì bị nhầm thành mẫu khác nhiều, P thấp vì nhiều mẫu khác bị nhầm sang neutral.

6 Biểu đồ Loss và Accuracy



Biểu đồ Loss:

- Train loss giảm đều -> mô hình học tốt trên tập huấn luyện
- Val loss giảm từ epoch 0 đến 3, sau đó tăng nhẹ trở lại từ epoch 4 đến 6

Dấu hiệu bắt đầu overfitting từ epoch 3 trở đi: vì

- Train loss tiếp tục giảm → mô hình học kỹ dữ liệu train.
- Val loss tăng → mô hình không còn tổng quát hóa tốt cho dữ liệu validation.

Biểu đồ Accuracy:

- Train Accuracy tăng liên tục -> học rất tốt trên tập huấn luyện

- Val Accuracy tăng nhanh giai đoạn đầu (epoch 0-2), sau đó dao động, gần như không tăng nhiều sau epoch 3

Nhận xét: Mô hình học tốt nhưng đang overfit nhẹ từ epoch 4 trở đi, early stopping nên dùng và dừng ở epoch 3-4 để tránh overfitting

7 Dự đoán cảm xúc cho 1 câu

Dự đoán cảm xúc cho 1 câu

```
✓ 0s
from tensorflow.keras.preprocessing.sequence import pad_sequences
import numpy as np

def predict_sentiment(sentence, tokenizer, model, max_len=200):
    sentence_cleaned = clean_text(sentence)
    sequence = tokenizer.texts_to_sequences([sentence_cleaned])
    padded = pad_sequences(sequence, maxlen=max_len, padding='post')
    prediction = model.predict(padded)
    label = np.argmax(prediction)

    label_map = {0: 'Negative', 1: 'Neutral', 2: 'Positive'}
    return label_map[label], prediction
```

- Hàm predict_sentiment: với câu đầu vào dạng chuỗi ký tự và được trả về một tuple gồm nhãn cảm xúc
- làm sạch dữ liệu đầu vào
- chuyển câu thành chuỗi số (tokenization)
- đệm chuỗi với padding
- và được dự đoán qua mô hình đã được huấn luyện từ trước.
- label = np.argmax(prediction): tìm nhãn có xác suất cao nhất để in ra

VD1:

```
sentence = "The food was absolutely wonderful and the service was excellent!"
label, confidence = predict_sentiment(sentence, tokenizer, model)
print("Sentiment:", label)
print("Confidence:", confidence)
```

```
1/1 ————— 0s 77ms/step
Sentiment: Positive
Confidence: [[0.0024343 0.01076291 0.98680276]]
```

Với câu đầu vào: "The food was absolutely wonderful and the service was excellent!" mang nghĩa tích cực mô hình đã dự đoán chính xác đúng như kỳ vọng. Ta thấy Confidence: [[0.0024343 0.01076291 0.98680276]] có giá trị lớn nhất là 0.98680276 tại chỉ số 2 -> positive

VD2:

```
[ ] sentence = "The food today is terrible."
    label, confidence = predict_sentiment(sentence, tokenizer, model)
    print("Sentiment:", label)
    print("Confidence:", confidence)
```

1/1 ————— 0s 41ms/step
Sentiment: Negative
Confidence: [[0.98087466 0.01058553 0.0085398]]

VD3:

```
[ ] sentence = "This dish was okay, but not great."
    label, confidence = predict_sentiment(sentence, tokenizer, model)
    print("Sentiment:", label)
    print("Confidence:", confidence)
```

1/1 ————— 0s 40ms/step
Sentiment: Neutral
Confidence: [[0.04091679 0.93796253 0.02112062]]

VI Hướng cải thiện

- Tăng chất lượng làm sạch văn bản. VD: giữ lại các từ phủ định như 'not', 'never', 'no' ...
- Tăng dữ liệu
- Xử lý mất cân bằng dữ liệu tốt hơn
- Cải thiện kiến trúc mô hình: tăng chất lượng embedding, tăng độ sâu, độ phức tạp,..
- Cải thiện huấn luyện mô hình

VII Kết luận

Bài tập lớn này đã thành công trong việc xây dựng một mô hình phân tích cảm xúc hiệu quả dựa trên Bi-LSTM, với khả năng xử lý dữ liệu văn bản và dự đoán cảm xúc (negative, neutral, positive) một cách chính xác. Các cải tiến được đề xuất (tăng chất lượng dữ liệu, tối ưu kiến trúc, tinh chỉnh huấn luyện) đã giúp mô hình đạt hiệu suất tốt hơn, đặc biệt trên các lớp thiểu số.

Mô hình hiện tại là một nền tảng vững chắc, nhưng có thể được nâng cấp thêm bằng cách chuyển sang các kiến trúc hiện đại (như BERT), tăng dữ

liệu đa dạng, và triển khai các kỹ thuật tối ưu hóa để đáp ứng các yêu cầu thực tế.

Link: https://colab.research.google.com/drive/1qgzD_8iScCpJctEdkusp=sharing