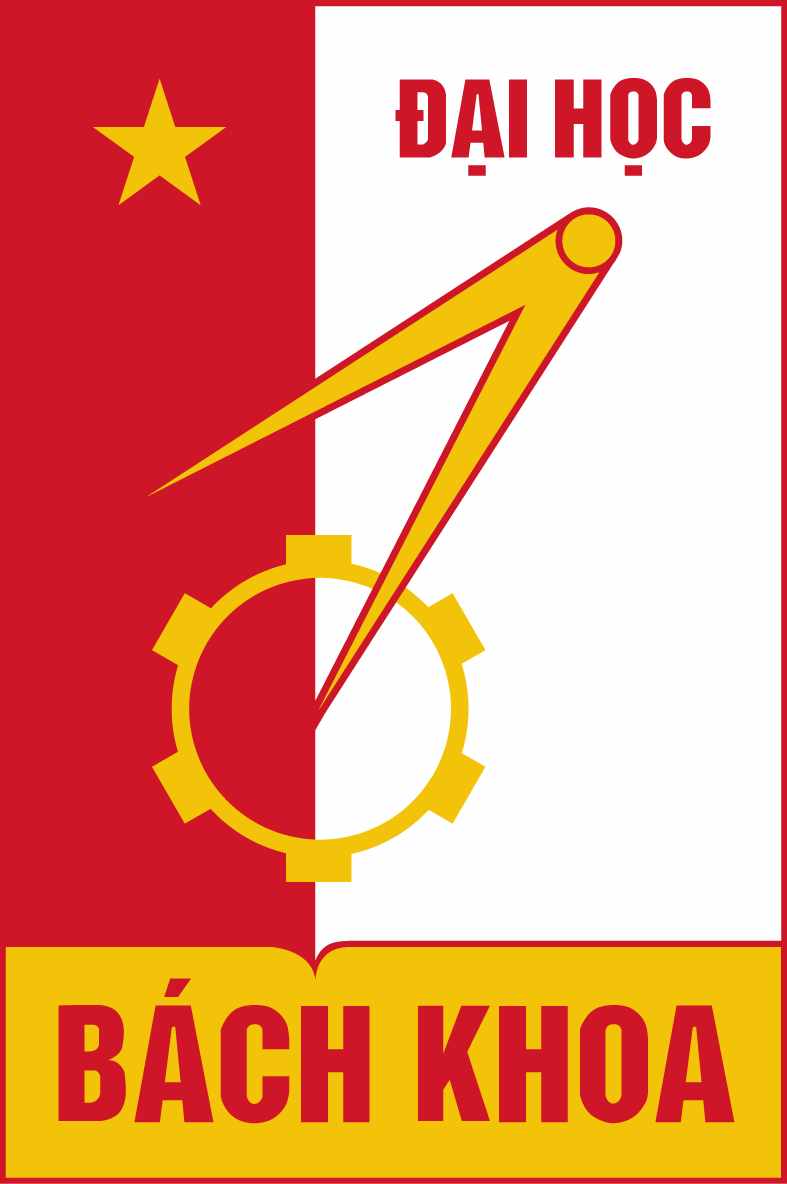
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG PHÂN LOẠI CẢM XÚC NGƯỜI DÙNG THÔNG QUA CÁC DÒNG TWEET**

**Sinh viên thực hiện:** Lê Anh Hào 20166042

Nguyễn Ngọc Nam 20162826

**Giáo viên hướng dẫn**: TS. Nguyễn Nhật Quang

 **Hà Nội 2020**

MỤC LỤC

[LỜI MỞ ĐẦU 3](#_Toc43790451)

[CHƯƠNG 1: GIỚI THIỆU KHÁI QUÁT VỀ MACHINE LEARNING VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN 4](#_Toc43790452)

[**1.1.** **Giới thiệu về Machine Learning.** 4](#_Toc43790453)

[**1.2. Sự phát triển của Deep learning trong nhưng năm gần đây.** 6](#_Toc43790454)

[**1.3. Xử lý ngôn ngữ tự nhiên** (**Natural Language Processing**). 7](#_Toc43790455)

[**1.3.1 Các bước xử lý.** 7](#_Toc43790456)

[**1.3.2** **Mô tả bài toán thực tế cần giải quyết** 7](#_Toc43790457)

[CHƯƠNG 2: CÁC BƯỚC GIẢI QUYẾT BÀI TOÁN PHÂN LOẠI VĂN BẢN 9](#_Toc43790458)

[**2.1. Mô hình phân loại văn bản.** 9](#_Toc43790459)

[**2.2. Bước làm sạch dữ liệu văn bản.** 9](#_Toc43790460)

[**2.3. Bước biến đổi từ thành véctơ (Word embedding).** 10](#_Toc43790461)

[**2.4. Sử dụng convolutional neural network để phân loại văn bản.** 14](#_Toc43790462)

[**2.4.1. Giới thiệu về mạng Nơ-ron tích chập.** 14](#_Toc43790463)

[**2.4.2. Kiến mạng nơ-ron tích chập trong phân loại cảm xúc văn bản**. 15](#_Toc43790464)

[CHƯƠNG 3: BÀI TOÁN THỰC TẾ VÀ CÁC KẾT QUẢ ĐẠT ĐƯỢC 21](#_Toc43790465)

[**3.1. Bài toán phân loại cảm xúc người dùng thông qua các dòng Tweet.** 21](#_Toc43790466)

[**3.1.1. Giới thiệu khái quát về bài toán.** 21](#_Toc43790467)

[**3.1.2. Bộ dữ liệu sẽ sử dụng.** 21](#_Toc43790468)

[**3.1.3. Các bước giải quyết bài toán.** 21](#_Toc43790469)

[**3.1.4. Các kết quả đạt được với mô hình đã xây dựng.** 24](#_Toc43790470)

[**3.2. Xây dựng hệ thống phân loại cảm xúc.** 25](#_Toc43790471)

[CHƯƠNG 4: KẾT LUẬN VÀ ĐỊNH HƯỚNG 28](#_Toc43790472)

[TÀI LIỆU THAM KHẢO 29](#_Toc43790473)

# ****LỜI MỞ ĐẦU****

**Cùng với xu thế toàn cầu hóa và hướng đến cách mạng công nghiệp 4.0 thì trí tuệ nhân tạo (AI) đang phát triển rất mạnh mẽ, đặc biệt là các vấn đề về Machine learning, deep learning đang được mọi người hết sức quan tâm dạo gần đây. Chúng được nhắc tới ở hầu hết các diễn đàn hay các hội thảo về công nghệ.**

Machine learning (học máy) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.Deep learning (học sâu) là một chi của ngành machine learning (máy học) dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Chúng em xin cảm ơn thầy giáo Nguyễn Nhật Quang, giảng viên bộ môn Trí tuệ nhân tạo, Viện CNTT&TT, trường ĐHBKHN đã tận tình hướng dẫn nhóm trong quá trình thực hiện báo cáo này. Thông qua quá trình tìm hiểu, tập hợp tài liệu và viết báo cáo, chúng em đã có cơ hội học hỏi và hiểu thêm về một số phương pháp phân loại cảm xúc văn bản.

Trong quá trình thực hiện, chúng em rất mong nhận được những góp ý của thầy giáo cũng như các bạn, nhằm cải thiện chất lượng bài báo cáo một cách tốt nhất!

Chúng em xin chân thành cảm ơn!

# ****CHƯƠNG 1: GIỚI THIỆU KHÁI QUÁT VỀ MACHINE LEARNING VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN****

## **Giới thiệu về Machine Learning.**

Machine Learning là một tập con của trí tuệ nhân tạo, nơi mà các thuật toán máy tính được sử dụng để tự học từ dữ liệu và thông tin. Trong machine learning, các máy tính không cần phải được lập trình một cách rõ ràng nhưng có thể tự thay đổi và cải thiện các thuật toán của chúng.

Hiện nay, các thuật toán machine learning cho phép máy tính có thể giao tiếp với con người, xe hơi tự lái, viết và xuất bản tường thuật các trận đấu thể thao, và tìm thấy kẻ tình nghi khủng bố. Tôi tin chắc rằng machine learning sẽ tác động một cách sâu sắc đến mọi ngành công nghiệp và các công việc liên quan đến chúng, đó là lý do tại sao mọi nhà quản lý cần phải có ít nhất một số kiến thức về machine learning và nó đã phát triển như thế nào.

Trong phần này tôi sẽ cung cấp một sơ lược theo thời gian về lịch sự hình thành của machine learning cũng như các sự kiện quan trọng gần đây nhất.

1950 - Nhà bác học Alan Turing đã tạo ra "Turing Test (phép thử Turing)" để xác định xem liệu một máy tính có trí thông minh thực sự hay không. Để vượt qua bài kiểm tra đó, một máy tính phải có khả năng đánh lừa một con người tin rằng nó cũng là con người. [1]

1952 - Arthur Samuel đã viết ra chương trình học máy (computer learning) đầu tiên. Chương trình này là trò chơi cờ đam, và hãng máy tính IBM đã cải tiến trò chơi này để nó có thể tự học và tổ chức những nước đi trong chiến lược để giành chiến thắng. [1]

1957 - Frank Rosenblatt đã thiết kế mạng nơron (neural network) đầu tiên cho máy tính, trong đó mô phỏng quá trình suy nghĩ của bộ não con người. [1]

1967 - Thuật toán "nearest neighbor" đã được viết, cho phép các máy tính bắt đầu sử dụng những mẫu nhận dạng (pattern recognition) rất cơ bản. Nó được sử dụng để vẽ ra lộ trình cho một người bán hàng có thể bắt đầu đi từ một thành phố ngẫu nhiên nhưng đảm bảo anh ta sẽ đi qua tất cả các thành phố khác theo một quãng đường ngắn nhất. [1]

1979 - Sinh viên tại trường đại học Stanford đã phát minh ra giỏ hàng "Stanford Cart" có thể điều hướng để tránh các chướng ngại vật trong một căn phòng. [1]

1981 - Gerald Dejong giới thiệu về khái niệm Explanation Based Learning (EBL), trong đó một máy tính phân tích dữ liệu huấn luyện và tạo ra một quy tắc chung để nó có thể làm theo bằng cách loại bỏ đi những dữ liệu không quan trọng. [1]

1985 - Terry Sejnowski đã phát minh ra NetTalk, nó có thể học cách phát âm các từ giống như cách một đứa trẻ tập nói. [1]

1990s - Machine Learning đã dịch chuyển từ cách tiếp cận hướng kiến thức (knowledge-driven) sang cách tiếp cận hướng dữ liệu (data-driven). Các nhà khoa học bắt đầu tạo ra các chương trình cho máy tính để phân tích một lượng lớn dữ liệu và rút ra các kết luận - hay là "học" từ các kết quả đó. [1]

1997 - Deep Blue của hãng IBM đã đánh bại nhà vô địch cờ vua thế giới. [1]

2006 - Geoffrey Hinton đã đưa ra một thuật ngữ "deep learning" để giải thích các thuật toán mới cho phép máy tính "nhìn thấy" và phân biệt các đối tượng và văn bản trong các hình ảnh và video. [1]

2010 - Microsoft Kinect có thể theo dõi 20 hành vi của con người ở một tốc độ 30 lần mỗi giây, cho phép con người tương tác với máy tính thông qua các hành động và cử chỉ. [1]

2011 - Máy tính Watson của hãng IBM đã đánh bại các đối thủ là con người tại Jeopardy. [1]

2011 - Google Brain đã được phát triển, và mạng deep nơron (deep neural network) của nó có thể học để phát hiện và phân loại nhiều đối tượng theo cách mà một con mèo thực hiện. [1]

2012 - X Lab của Google phát triển một thuật toán machine learning có khả năng tự động duyệt qua các video trên YouTube để xác định xem video nào có chứa những con mèo. [1]

2014 - Facebook phát triển DeepFace, một phần mềm thuật toán có thể nhận dạng hoặc xác minh các cá nhân dựa vào hình ảnh ở mức độ giống như con người có thể. [1]

2015 - Amazon ra mắt nền tảng machine learning riêng của mình. [1]

2015 - Microsoft tạo ra Distributed Machine Learning Toolkit, trong đó cho phép phân phối hiệu quả các vấn đề machine learning trên nhiều máy tính. [1]

2015 - Hơn 3.000 nhà nghiên cứu AI và Robotics, được sự ủng hộ bởi những nhà khoa học nổi tiếng như Stephen Hawking, Elon Musk và Steve Wozniak (và nhiều người khác), đã ký vào một bức thư ngỏ để cảnh báo về sự nguy hiểm của vũ khí tự động trong việc lựa chọn và tham gia vào các mục tiêu mà không có sự can thiệp của con người. [1]

2016 - Thuật toán trí tuệ nhân tạo của Google đã đánh bại nhà vô địch trò chơi Cờ Vây, được cho là trò chơi phức tạp nhất thế giới (khó hơn trò chơi cờ vua rất nhiều). Thuật toán AlphaGo được phát triển bởi Google DeepMind đã giành chiến thắng 4/5 trước nhà vô địch Cờ Vây. [1]

## **1.2. Sự phát triển của Deep learning trong nhưng năm gần đây.**

**Một trong những nguyên nhân chính dẫn đến sự phát triển của AI và machine learning chính là sự xuất hiện của deep learning.**

**Có rất nhiều nguyên nhân dẫn đến sự phát triển và thành công của machine learning:**

* Sự ra đời của các bộ dữ liệu lớn được gán nhãn.
* Khả năng tính toán song song tốc độ cao của GPU.
* Sự ra đời của ReLU và các hàm kích hoạt liên quan làm hạn chế vấn đề vanishing gradient.
* Sự cải tiến của các kiến trúc: GoogLeNet, VGG, ResNet, … và các kỹ thuật transfer learning, fine tuning.
* Nhiều kỹ thuật regularization mới: dropout, batch normalization, data augmentation.
* Nhiều thư viện mới hỗ trợ việc huấn luyện deep network với GPU: theano, caffe, mxnet, tensorflow, pytorch, keras, …
* Nhiều kỹ thuật tối ưu mới: Adagrad, RMSProp, Adam, …

## **1.3. Xử lý ngôn ngữ tự nhiên (Natural Language Processing).**

NLP (Natural Language Processing) là khái niệm để chỉ các kĩ thuật, phương pháp thao tác trên ngôn ngữ tự nhiên bằng máy tính. Bạn cần phân biệt ngôn ngữ tự nhiên (ví dụ như tiếng Việt, tiếng Anh, tiếng Nhật… là những ngôn ngữ trong giao tiếp thường ngày) và ngôn ngữ nhân tạo (như ngôn ngữ lập trình, ngôn ngữ máy, …).

Trong NLP có 2 quan điểm cơ bản:

* Xử lý các từ ngữ bằng máy tính.
* Làm cho máy tính hiểu được các từ ngữ.

Hiện tại, cả 2 hướng này đều đang được tích cực nghiên cứu và phát triển, nhờ đó rất nhiều các hệ thống hiệu quả đã và đang được tạo ra.

### **1.3.1 Các bước xử lý.**

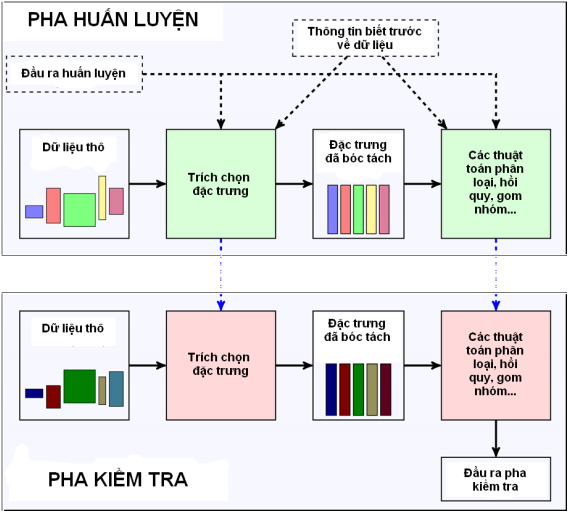
* **Phân tích hình thái** - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong tiếng Anh và nhiều ngôn ngữ khác, các từ được phân tách với nhau bằng dấu cách. Tuy nhiên trong tiếng Việt, dấu cách được dùng để phân tách các tiếng (âm tiết) chứ không phải từ. Cùng với các ngôn ngữ như tiếng Trung, tiếng Hàn, tiếng Nhật, phân tách từ trong tiếng Việt là một công việc không hề đơn giản.
* **Phân tích cú pháp** - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.
* **Phân tích ngữ nghĩa** - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.
* **Tích hợp văn bản** - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.
* **Phân tích thực nghĩa** - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

Tuy nhiên, ranh giới giữa 5 bước xử lý này cũng rất mong manh. Chúng có thể được tiến hành từng bước một, hoặc tiến hành cùng lúc - tùy thuộc vào giải thuật và ngữ cảnh cụ thể.

* + 1. **Mô tả bài toán thực tế cần giải quyết**
* Yêu cầu của bài toán : Ta có dòng tweet trên twitter của các người dùng khác nhau. Yêu cầu đặt ra là phải phát hiện thái độ cảm xúc của người viết các dòng tweet đó. Với một dòng tweet yêu cầu máy phải đưa ra được thái độ cảm xúc của người viết dòng tweet đó là tích cực (positive) hay tiêu cực (negative).
* Yêu cầu cần đạt được : Xây dựng được một mô hình có thể phát hiện được cảm xúc của người viết thông qua các dòng tweet trên twitter (negative, neutral)
* Kịch bản ứng dụng : Ta chọn một file text chứa nội dung của một dòng tweet. Hệ thống sẽ trả lời dòng tweet đó mang sắc thái biểu cảm gì (negative, positive).

# CHƯƠNG 2: CÁC BƯỚC GIẢI QUYẾT BÀI TOÁN PHÂN LOẠI VĂN BẢN

## **2.1. Mô hình phân loại văn bản.**

****

Hình 2.1 Mô hình chung cho các bài toán Machine learning

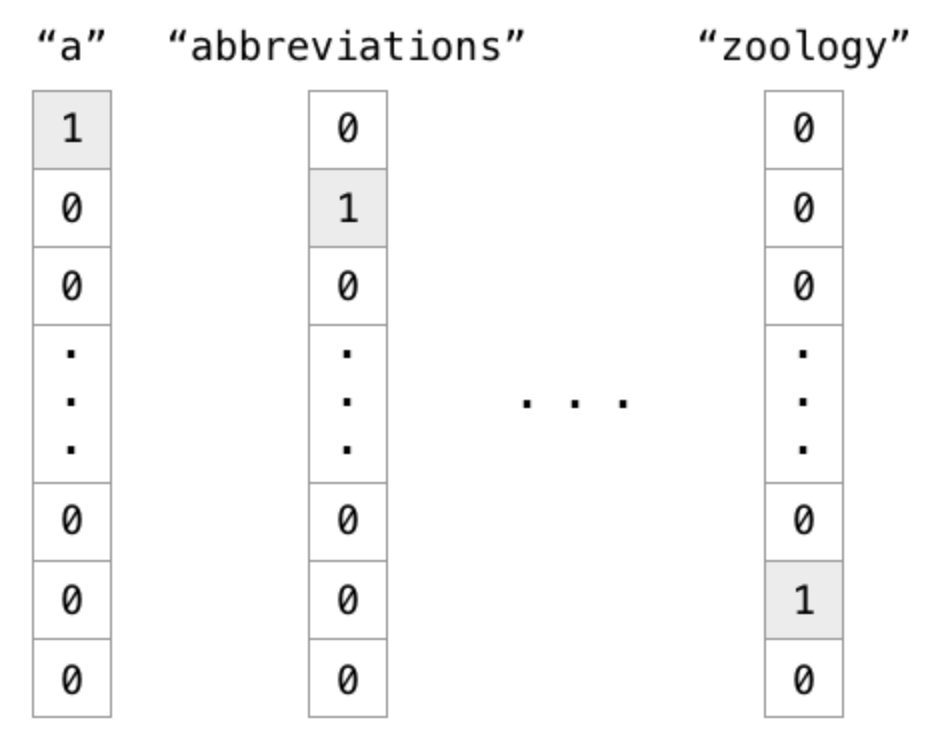
* **Hầu hết các mô hình phân loại văn bản đều có các bước như mô hình trên. Nó cũng là mô hình chung cho đại đa số các bài toán Machine learning.**
* **Theo mô hình trên thì phần dữ liệu đầu vào sẽ là các dòng tweet trên twitter. Kết quả đầu ra sẽ là phân loại dòng tweet này theo 2 nhãn negative và positive.**
* **Đối với bài toán phân loại cảm xúc văn bản – một bài toán thuộc lĩnh vực** NLP (Natural Language Processing) thì việc xử lý dữ liệu thô và trích chọn đặc trưng của dữ liệu là việc ta đi làm sạch dữ liệu, biến đổi từ thành vectơ,…

## **2.2. Bước làm sạch dữ liệu văn bản.**

* **Đầu tiên ta phải xóa bỏ các kí hiệu đặc không có tác dụng về ngữ nghĩa như các dấu câu hay các ký hiệu @, #, $, %…**
* **Tiếp theo, chúng ta phải chuẩn hóa chữ viết, biến các chữ viết hoa thành các chữ thường đưa các chữ viết về cùng một loại. Ví dụ: ‘Class’ sẽ thành ‘class’ điều này sẽ giúp máy nhân biết 2 chữ ‘Class’ và ‘class’ là giống nhau do một từ ở đầu câu được viết hoa.**
* **Sau đó, ta phải loại bỏ các từ không mang nhiều ý nghĩa nhưng lại xuất hiên rất nhiều trong các văn bản như ‘là’, ‘đây’, ‘đó’, ‘này’…Việc loại bỏ các từ này giúp việc train sẽ tạo ra một model tốt. Những từ được loại bỏ này được gọi là stopwords.**
* **Cuối cùng, ta phải tách các từ trong một câu ra điều này nhàm mục đích biến đổi một từ thành một véctơ và khi đó một văn bản sẽ là ma trận được tạo thành khi ghép các véc tơ của các từ lại. Ta có thể tách 1 từ 2 từ hay 3 từ tùy vào từng trường hợp cụ thể để có thể có một tập dữ đầu vào tốt phù hợp với các model phân loại.**

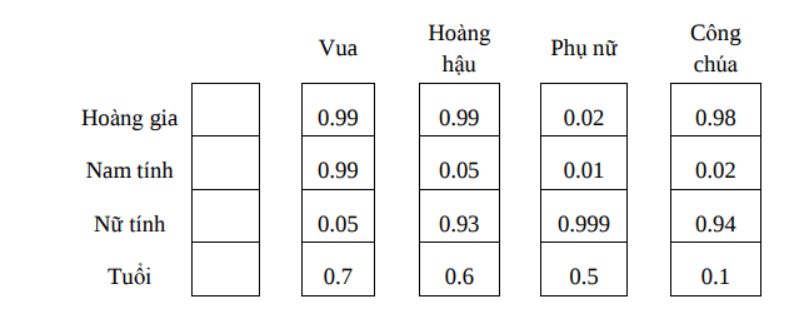
### **2.3. Bước biến đổi từ thành véctơ (Word embedding).**

* Word embedding là một nhóm các kỹ thuật đặc biệt trong xử lý ngôn ngữ tự nhiên, có nhiệm vụ ánh xạ một từ hoặc một cụm từ trong bộ từ vựng tới một vector số thực. Từ không gian một chiều cho mỗi từ tới không gian các vector liên tục. Các vector từ được biểu diễn theo phương pháp word embedding thể hiện được ngữ nghĩa của các từ, từ đó ta có thể nhận ra được mối quan hệ giữa các từ với nhau (tương đồng, trái nghịch, ...).
* Trong các ứng dụng về xử lý ngôn ngữ tự nhiên, học máy, ... các thuật toán không thể nhận được đầu vào là chữ với dạng biểu diễn thông thường. Để máy tính có thể hiểu được, ta cần chuyển các từ trong ngôn ngữ tự nhiên về dạng mà các thuật toán có thể hiểu được (dạng số). Cách truyền thống để biểu diễn một từ là dùng one-hot vector. Ta thực hiên các bước như sau:
* Xây dựng một bộ từ vựng.
* Mỗi vector đại diện cho một từ có số chiều bằng số từ trong bộ từ vựng. Trong đó, mỗi vector chỉ có một phần tử duy nhất khác 0(bằng 1) tại vị trí tương ứng với vị trí từ đó trong bộ từ vựng.

****

Hình 2.2 Biểu diễn cách biến đổi một từ thành véctơ sử dụng one-hot vector.

* Tuy nhiên phương pháp này lại có những nhược điểm cụ thể là: độ dài của vectơ là quá lớn khi kích thước bộ từ vựng tăng, không thể hiện được mối quan hệ giữa các từ và tính tương đồng. Phương pháp được đưa ra để giải quyết vấn đề trên là Word2vec.
* Thay vì đếm và xây dựng ma trận đồng xuất hiện, word2vec học trực tiếp word vector có số chiều thấp trong quá trình dự đoán các từ xung quanh mỗi từ. Đặc điểm của phương pháp này là nhanh hơn và có thể dễ dàng kết hợp một câu một văn bản mới hoặc thêm vào từ vựng.
* Word2vec là một mạng neural 2 lớp với duy nhất 1 tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector (với số chiều khoảng vài trăm), với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian.
* Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian. Dự đoán chính xác cao về ý nghĩa của một từ dựa trên những lần xuất hiện trước đây.
* Nếu ta gán nhãn các thuộc tính cho một vector từ giả thiết, thì các vector được biểu diễn theo word2vec sẽ có dạng như sau:



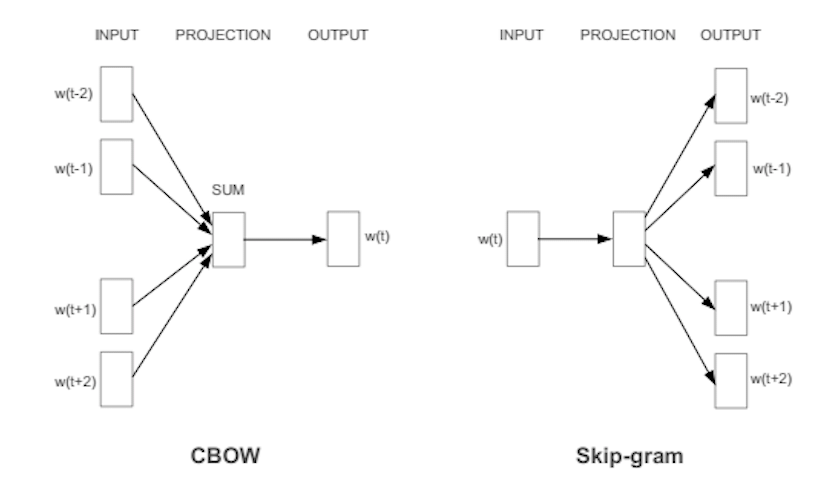
Hình 2.3 Biểu diễn cách biến đổi một từ thành véctơ sử dụng Word2vec

* Có 2 mô hình Word2vec được áp dụng: Skip-gram, Continuous Bag of Words (CBOW).
* ***Skip-gram****:*

Input là từ cần tìm mối quan hệ, output là từ các từ có quan hệ gần nhất với từ đó. Ví dụ câu: "I have a cute dog", input từ "a", output là “I”, “have”, “cute”, “dog”.

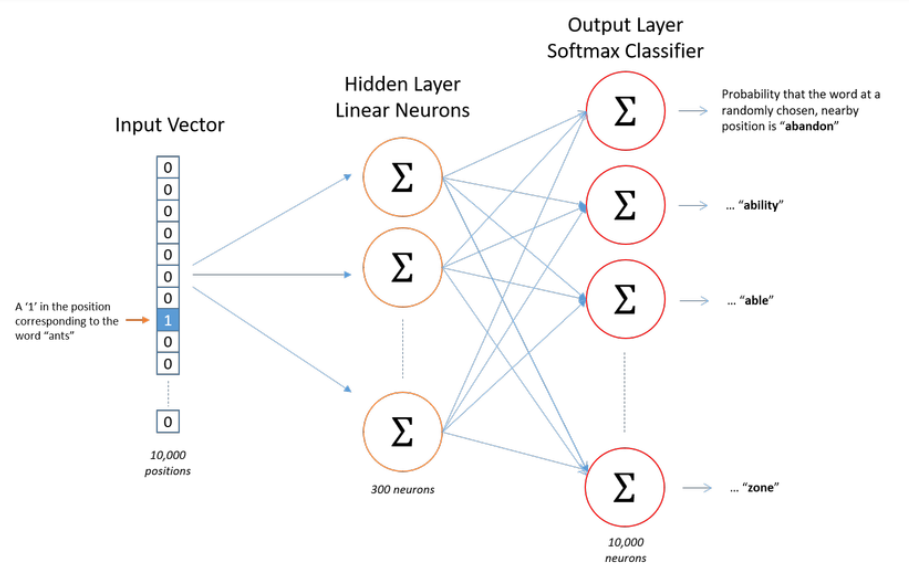
* ***Continuous Bag of Words:***

CBOW thì ngược lại, input context, output là từ gần nhất với context đó.



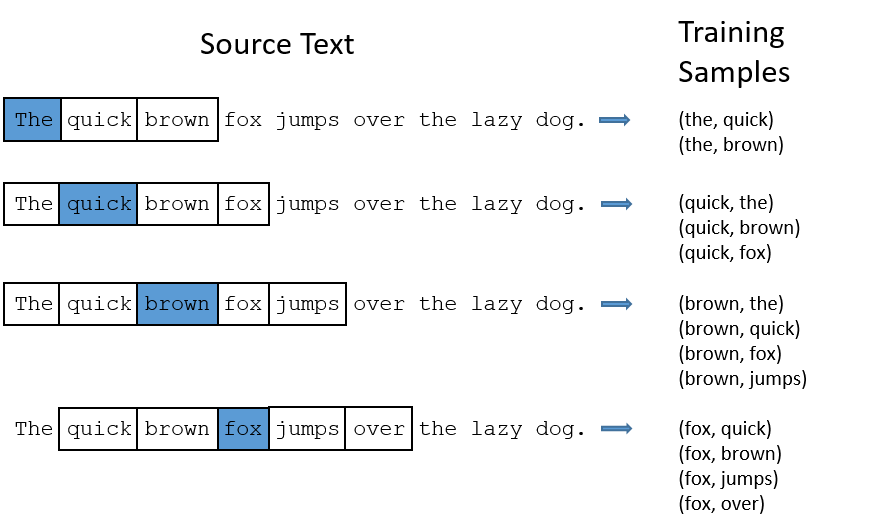
Hình 2.4 Mô hình CBOW và Skip-gram

* **Do *skip-gram* cho kết quả tốt hơn với dữ liệu lớn nên ta thường sử dụng nó.**
* **Mục tiêu**: Học trọng số các lớp ẩn, các trọng số này là các words vector
* **Cách thức**: Cho một từ cụ thể ở giữa câu (input word), nhìn vào những từ ở gần và chọn ngẫu nhiên. Mạng neural sẽ cho chúng ta biết xác suất của mỗi từ trong từ vựng về việc trở thành từ gần đó mà chúng ta chọn.
* Dưới đây là mô hình kiến trúc của mạng Skip-gram và cách xây dựng training data:



**Hình 2.5** Mô hình kiến trúc của mạng Skip-gram

* **Mô hình trainning:** Xây dựng training data với windows size = 2. Ở đây windows được hiểu như một cửa sổ trượt qua mỗi từ. Windows size = 2 tức là lấy 2 từ bên trái và bên phải mỗi từ trung tâm.



**Hình 2.6 Ví dụ về mô hình training**

* Model detail:
* Xây dựng bộ từ vựng
* Biểu diễn mỗi từ thành các one-hot-vector
* Đầu ra là một vector duy nhất, có kích thước bằng kích thước của bộ từ vựng, thể hiện xác suất của mỗi từ được là lân cận của từ đầu vào.
* Không có hàm kích hoạt trên tầng ẩn
* Hàm kích hoạt trên tầng output là softmax
* Trong quá trình training, input là 1 one-hotvector, ouput cũng là 1 one-hot-vector
* Trong quá trình đánh giá sau khi training, đầu ra phải là 1 phân bố xác suất.

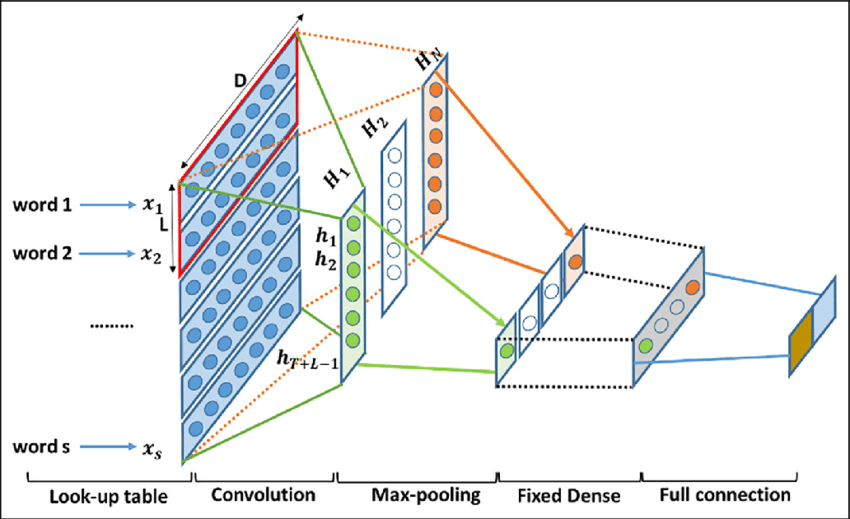
## **2.4. Sử dụng convolutional neural network để phân loại văn bản.**

### **2.4.1. Giới thiệu về mạng Nơ-ron tích chập.**

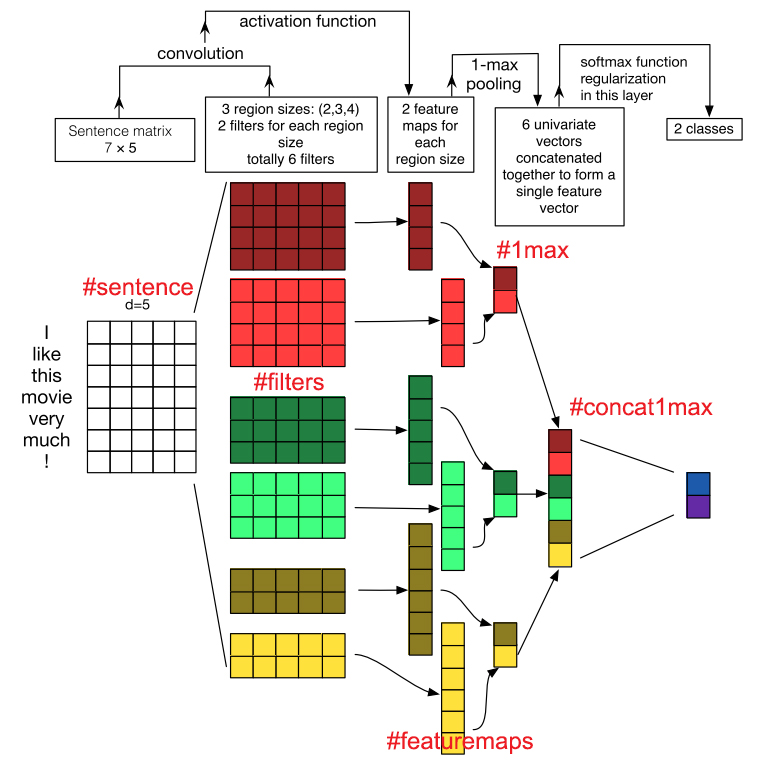
* Mạng nơ-ron tích chập (CNN - Convolutional Neural Network) là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớnvcủa Facebook, Google hay Amazon… cho các mục đích khác nhau như các thuật toán tagging tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.
* Sự ra đời của mạng CNN là dựa trên ý tưởng cải tiến cách thức các mạng nơ-ronnhân tạo truyền thống học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng nơ-ron nhân tạo truyền thẳng (Feedforward NeuralNetwork) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh và kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra sự liên kết đầy đủ này cũng là sự dư thừa khi với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. Mạng CNN ra đời với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ-ron truyền thẳng.

### **2.4.2. Kiến mạng nơ-ron tích chập trong phân loại cảm xúc văn bản**.

Các lớp cơ bản trong một mạng CNN bao gồm: Lớp tích chập (Convolutional), Lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit), Lớp lấy mẫu (Pooling) và Lớp kết nối đầy đủ (Fully-connected), được thay đổi về số lượng và cách sắp xếp để tạo ra các mô hình huấn luyện phù hợp cho từng bài toán khác nhau.



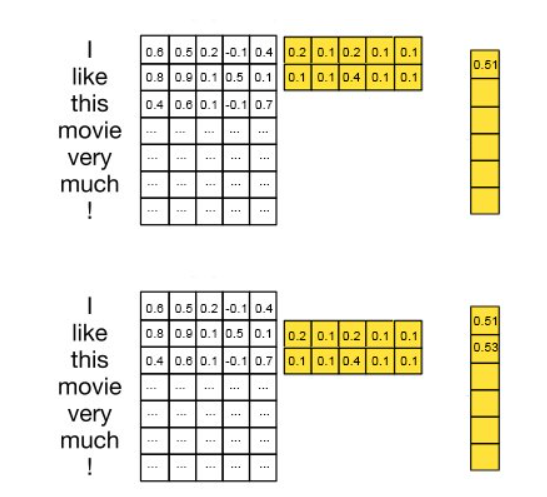
Hình 2.7 Kiến trúc cơ bản của một mạng tích chập về phân loại văn bản.



Hình 2.8 Ví dụ về một mô hình CNN phân loại văn bản.

* **Lớp tích chập:**

Đây là thành phần quan trọng nhất trong mạng CNN, cũng là nơi thể hiệntư tưởng xây dựng sự liên kết cục bộ thay vì kết nối toàn bộ các từ. Các liên kết cục bộ này được tính toán bằng phép tích chập giữa ma trận nhúng của một câu với các bộ lọc – filters – có kích thước cố định một chiều bằng số chiều của wordvectơ, khác so với khi tích chập đối với ảnh.



**Hình 2.9 Ví dụ tính tích chập đối với bộ lọc kích thước là 2.**

Trong ví dụ ở Hình 2.9, ta thấy ma trận đầu vào có kích thước là 7x5 tương ứng với 1 câu có 7 từ mỗi từ được biến đổi thành một vectơ 5 chiều, bộ lọc được sử dụng là một ma trận có kích thước 2x5. Bộ lọc này được dịch chuyển lần lượt qua từng từ đến khi hoàn thành quét toàn bộ câu, tạo ra một ma trận mới có kích thước 6x1 nhỏ hơn kích thước ma trận đầu vào.

* **Lớp kích hoạt phi tuyến ReLU**:

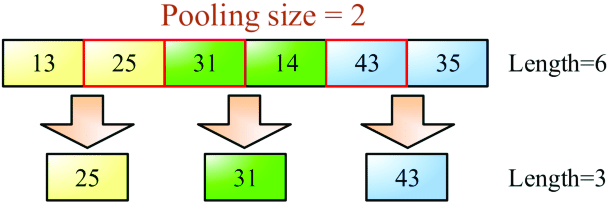
Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp Tích chập. Lớp Kích hoạt phi tuyến nói chung sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh… để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Cụ thể, phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.

f(x) = max (0, x)

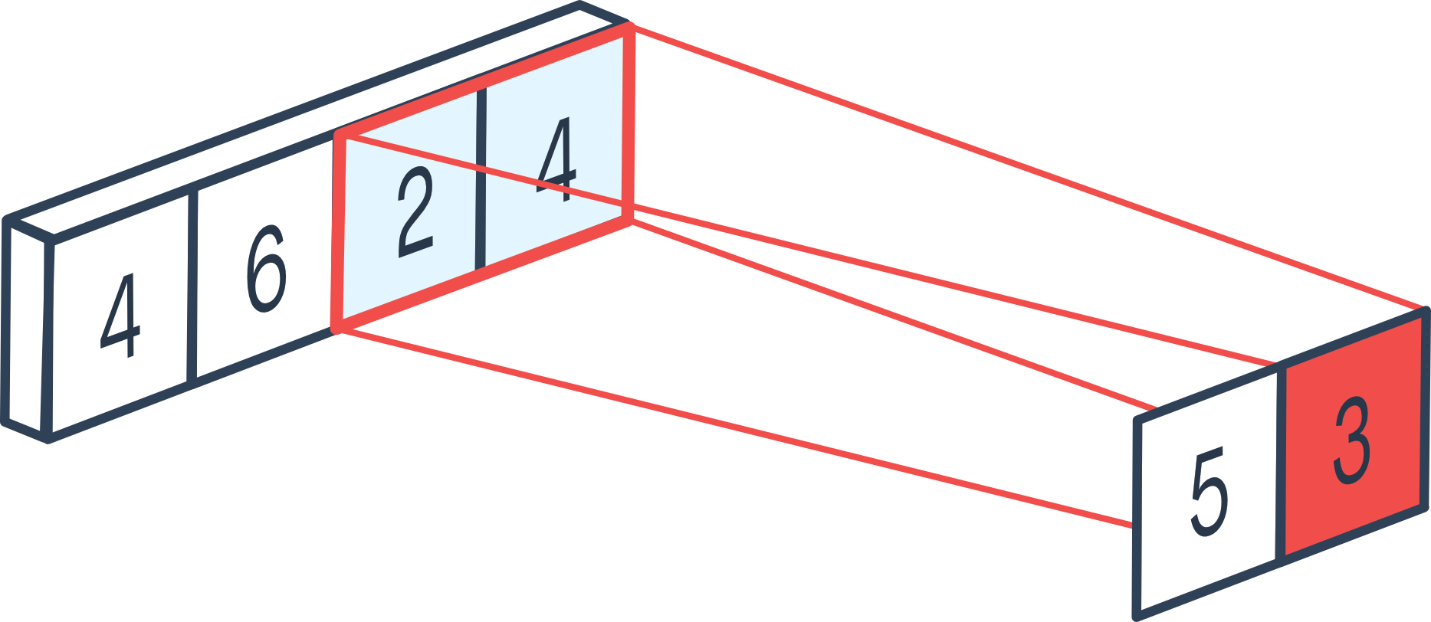
Thông thường, lớp ReLU được áp dụng ngay phía sau lớp Tích chập, với đầu ra là một ma trận mới có kích thước giống với ma trận đầu vào, các giá trị của điểm trên ma trận cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.

* **Lớp lấy mẫu:**

Một thành phần tính toán chính khác trong mạng CNN là lấy mẫu (Pooling),  
thường được đặt sau lớp Tích chập và lớp ReLU để làm giảm kích thước kích thước ma trận đầu ra trong khi vẫn giữ được các thông tin quan trọng của ma trận đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các đăc trưng vừa được lấy ra từ lớp tích chập và thực hiện phép lấy mẫu thay vì phép tích chập – tức là ta sẽ chọn lưu lại một giá trị đặc trưng tiêu biểu nhất trong các đặc trưng ta vừa quét. Có 2 cách lấy mẫu được sử dụng nhiều nhất hiện nay đó là Max Pooling (lấy giá trị của điểm lớn nhất) và Avarage Pooling (lấy giá trị trung bình của các điểm trên các đặc trưng).



**Hình 2.10 Phương pháp Max Pooling**



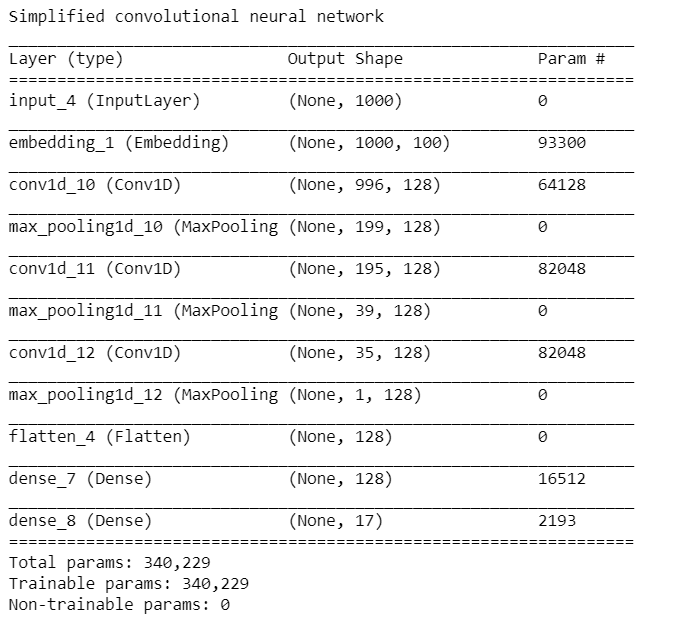
**Hình 2.11 Phương pháp** Avarage Pooling

Như vậy, với mỗi ma trận đặc trưng được đưa qua lấy mẫu ta thu được một ma trận đăc trưng tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán sau này.

* **Lớp kết nối đầy đủ:**

Lớp kết nối đầy đủ này được thiết kế hoàn toàn tương tự như trong mạng nơ-ron truyền thống, tức là tất cả các điểm trên ma trận được kết nối đầy đủ với node trong lớp tiếp theo. So với mạng nơ-ron truyền thống, các ma trận đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ ron truyền thống.

* Các lớp trên kết hợp thành một mô hình Convolutional neural network được sử dụng rất thường xuyên trong các bài toán phân loại. Nó cũng được ứng dụng trong bài toán phân loại cảm xúc văn bản. Dưới đây là một mô hình phân loại văn bản sử dụng Convolutional neural network (CNN).



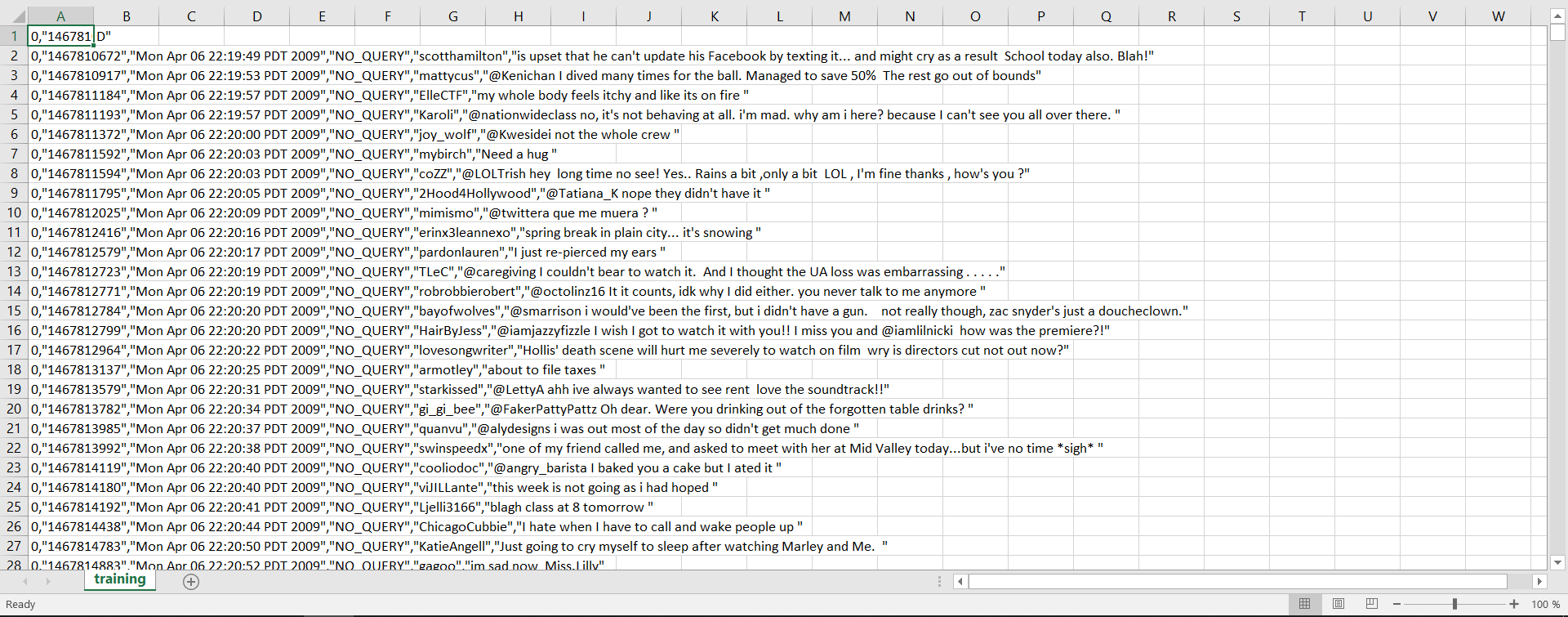
**Hình 2.12** Ví dụ về một mô hình Convolutional neural network sử dụng phân loại văn bản

# CHƯƠNG 3: BÀI TOÁN THỰC TẾ VÀ CÁC KẾT QUẢ ĐẠT ĐƯỢC

## **3.1. Bài toán phân loại cảm xúc người dùng thông qua các dòng Tweet.**

### **3.1.1. Giới thiệu khái quát về bài toán.**

* Yêu cầu của bài toán : Ta có dòng tweet trên twitter của các người dùng khác nhau. Yêu cầu đặt ra là phải phát hiện thái độ cảm xúc của người viết các dòng tweet đó. Với một dòng tweet yêu cầu máy phải đưa ra được thái độ cảm xúc của người viết dòng tweet đó là tích cực (positive) hay tiêu cực (negative).
* Input đầu vào : file .txt hoặc file .csv chứa nội dung các dòng tweet.



Hình 3.1 Dữ liệu đầu vào

* Output đầu ra : sắc thái của các dòng Tweet (Positive hay negative).

### **3.1.2. Bộ dữ liệu sẽ sử dụng.**

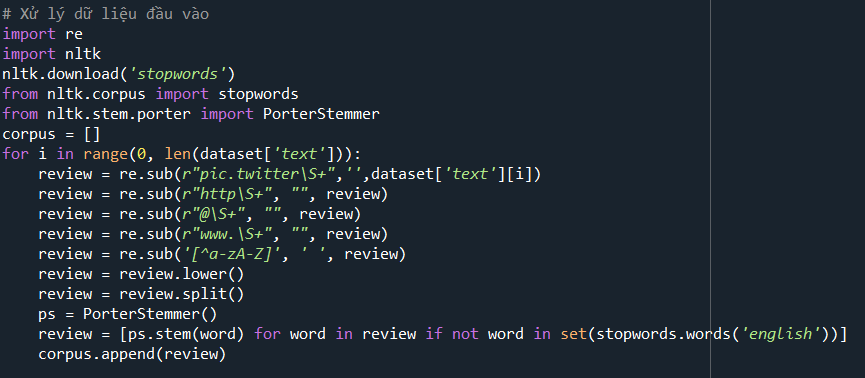
* Bộ dữ liệu được lấy từ nguồn : <https://www.kaggle.com/kazanova/sentiment140>

Bao gồm 1600000 dòng Tweet lấy trên twiter đã được gán sẵn nhãn.

* Ta chia bộ dữ liệu làm 3 tập : tập trainning, tập validation, tập test theo tỉ lệ 3 :1 :1
* Ta sẽ dùng tập training và tập validation để xây dựng mô hình và tối ưu các tham số của mô hình.
* Ta sẽ dùng tập test để kiểm tra mô hình vừa xây dựng.

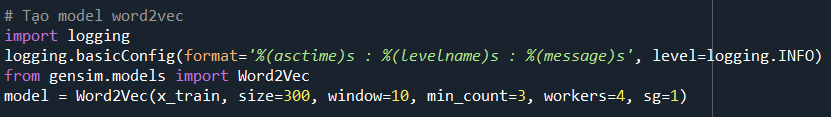
### **3.1.3. Các bước giải quyết bài toán.**

* Làm sạch dữ liệu :
* Loại bỏ các từ xuất hiện nhiều nhưng không mang ý nghĩa (stopwords), ta sử dụng stopwords có sẵn trong thư viện ntlk. Ví dụ về stopwords : is, a, am…
* Loại bỏ các cụm từ đặc biệt, các kí tự đặc biệt trong câu. Ví dụ như : <http://twitpic.com/2y1zl>, @switchfoot, pic.twitter, www hay các dấu câu như chấm, hỏi chấm, ngạch ngang,… Ta sử dụng thư viện re để xử lý chuỗi kí tự.
* Tách từ, biến đổi các chữ hoa thành chữ thường, chuẩn hóa từ. Ví dụ : ‘I have five books’ ta sẽ chuyển thành ‘i’, ‘have’, ‘five’, ‘book’. Chữ ‘I’ thành ‘i’ và ‘books’ thành ‘book’. Ta dùng thư viện ntlk để xử lý.



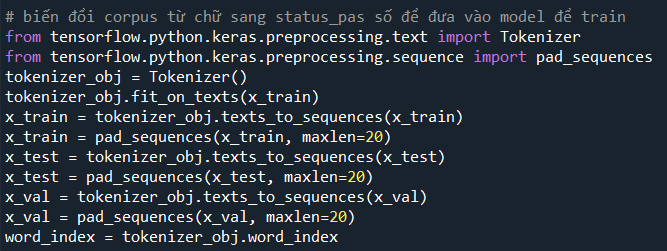
Hình 3.2 Code làm sạch dữ liệu.

* Biến đổi từ thành véctơ sử dung model Word2vec : Ta sử dụng thư viện gensim

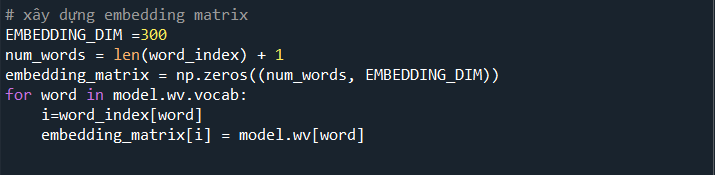


Hình 3.3 Code sử dụng model Word2vec.

* Xây dựng bộ từ vụng từ tập training, biến đổi các từ thành số, xây dựng ma trận nhúng (embedding matrix). Ta sử dụng thư viên keras.

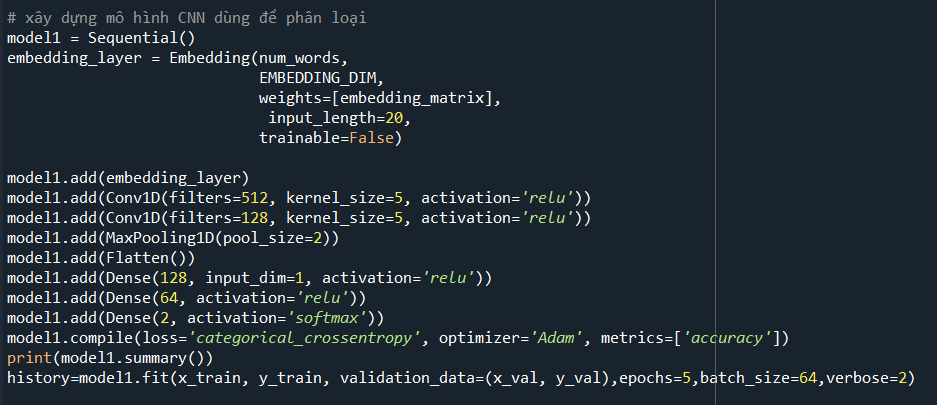


Hình 3.4 Code biến đổi sequences từ chữ thành số.

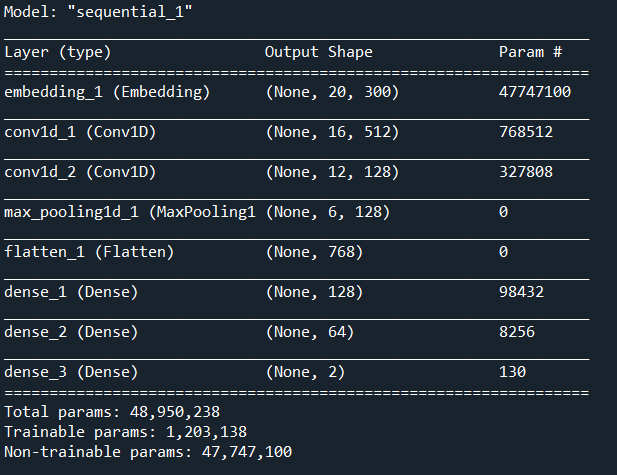


Hình 3.5 Code xây dựng ma trận nhúng.

* Xây dựng mô hình **Convolutional neural network và đưa dữ liệu vào train. Ta sử dụng thư viện keras.**

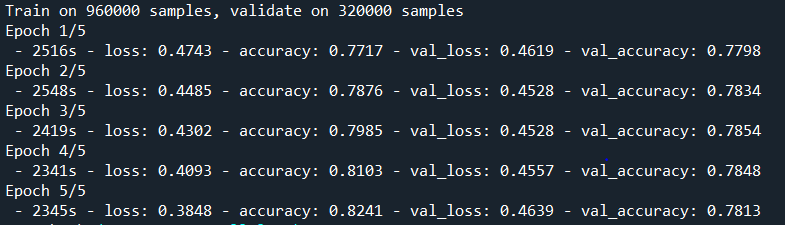


**Hình 3.6 Code xây dựng mô hình CNN và đưa dữ liệu vào train.**

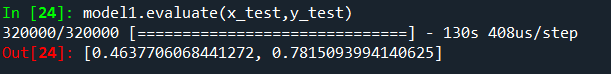
****

**Hình 3.7 Kiến trúc mô hình CNN được xây dựng**

### **3.1.4. Các kết quả đạt được với mô hình đã xây dựng.**

****

**Hình 3.8 Kết quả train với mô hình đã xây dựng**

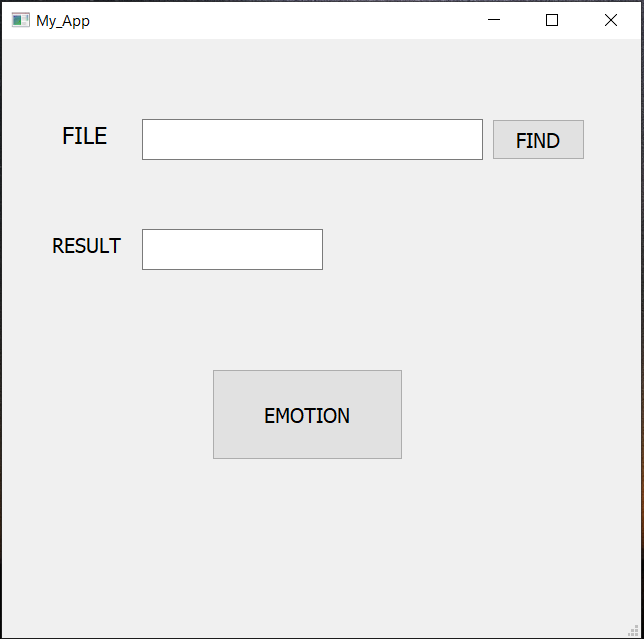
****

**Hình 3.9 Kết quả khi kiểm tra model trên tập test**

**Nhận xét : Ta thấy độ chính xác của model trên tập train và validation là 0.8241 và 0.7813. Đô chính xác của model mà ta xây dựng là không cao lắm. Ta thấy độ chính xác của model đo trên tập test là 0.7815 như vậy model mà ta xây dựng là chấp nhận được (ổn định) do kết quả đo trên tập train, tập validation và tập test là gần tương đương nhau cho thấy model ta xây dựng không bị underfitting và overfitting.**

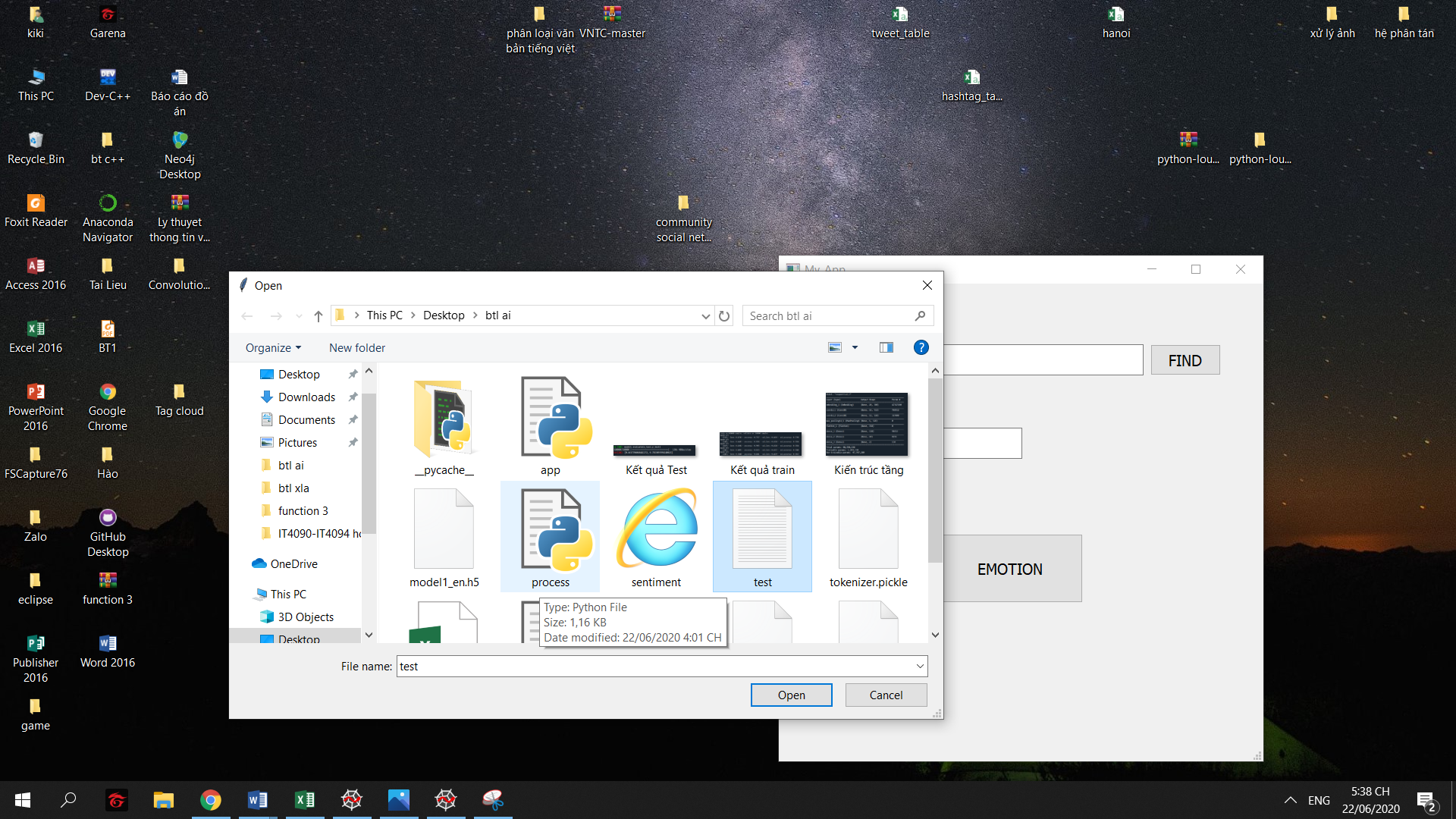
## **3.2. Xây dựng hệ thống phân loại cảm xúc.**

* **Giao diện của hệ thống :**

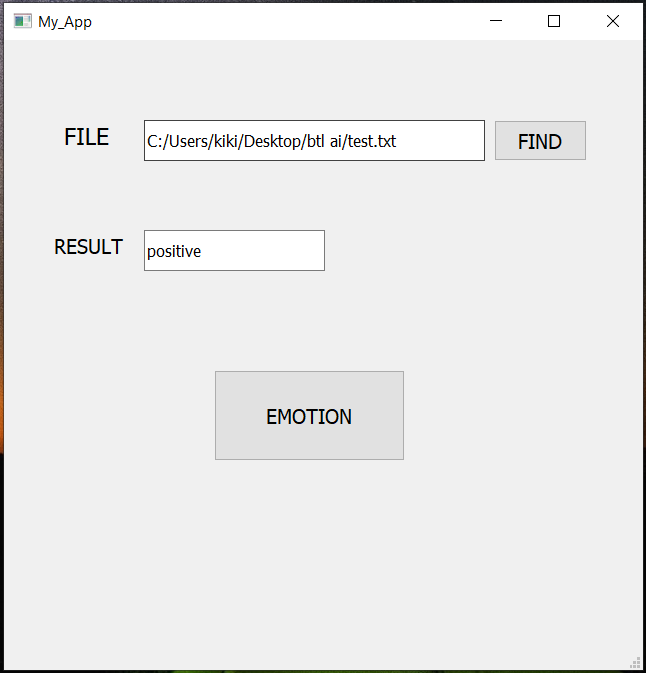


**Hình 3.10 Giao diện hệ thống**

* **Chức năng của hệ thống là phân loại cảm xúc của một dòng tweet.**
* **Cách sử dụng : ta copy đường dẫn của một file .txt vào mục FILE hoặc chọn tìm kiếm file .txt trên máy nhấn vào FIND, sau đó nhấn EMOTION ta sẽ có kết quả hiển thị trên RESULT.**



**Hình 3.11 Chọn file từ máy**



**Hình 3.12 Kết quả trả về cảm xúc dòng tweet trong file test.txt**

# ****CHƯƠNG 4: KẾT LUẬN VÀ ĐỊNH HƯỚNG****

**Như vậy, qua bài tập lớn nhóm em đã có thêm được nhiều kiến thức cơ bản về xử lý ngôn ngữ tự nhiên và các mô hình học máy phổ biến hiện nay. Trong quá trình tìm hiểu nhóm có gặp một số vấn đề về lý thuyết và phải tìm hiểu các thư viện để có thể sử dụng tương đối khó do các thành viên trong nhóm có trình độ ngoại ngữ không tốt mà các tài liệu về thư viện đều viết bằng tiếng anh. Tuy nhiên, nhóm đã cố gắng để hoàn thành cơ bản những gì đã đặt ra dù vẫn còn nhiều thiếu sót. Việc xử lý và phân loại cảm xúc văn bản là rất cần thiết trong xã hội hiện đại ngày nay. Nó có thể giúp chúng ta phân loại các bình luận đánh giá sản phẩm của khách hàng để từ đó biết được một sản phâm này có được ưu thích trên thị trường hay không từ đó có thể đưa ra phương án kinh doanh hợp lý, có thể tiếp thị những sản phẩm tương đồng đến những khách hàng đưa ra bình luận tích cực. Điều này rất có lợi với các công ty kinh doanh. Bài toán phân loại cảm xúc người dùng thông qua các văn bản thực chất là bài toán phân loại văn bản với 2 nhãn là negative và positive ta có thể mở rộng ra bài toán phân loại văn bản theo chủ đề với nhiều nhãn khác nhau như thể thao, du lịch, xã hội… Ngoài ra còn rất nhiều ứng dụng khác mà xử lý ngôn ngữ tự nhiên và học máy mang lại. Chính vì vậy mà xử lý ngôn ngư tự nhiên và các mô hình học máy ngày nay rất được quan tâm.**

# ****TÀI LIỆU THAM KHẢO****

[1] <https://techmaster.vn/posts/33923/lich-su-phat-trien-machine-learning>

[2] <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>

[3] <https://viblo.asia/newest>

[4] <https://viblo.asia/p/so-luoc-word2vec-6J3ZgBoAKmB>

[5] [Machine Learning trong Xử Lý Ngôn Ngữ Tự Nhiên - Nhóm Đông Du Nhật Bản](http://viet.jnlp.org/kien-thuc-co-ban-ve-xu-ly-ngon-ngu-tu-nhien/machine-learning-trong-nlp)

<http://viet.jnlp.org/kien-thuc-co-ban-ve-xu-ly-ngon-ngu-tu-nhien/machine-learning-trong-nlp>

**[6] Machine learning cơ bản – Vũ Hữu Tiệp**

<https://machinelearningcoban.com/>

[7] <https://www.kaggle.com/kazanova/sentiment140>

[8] <https://github.com/jatana-research/Text-Classification/blob/master/CNN.ipynb>

[9] <https://nttuan8.com/>

[10]<https://viblo.asia/p/xay-dung-mo-hinh-khong-gian-vector-cho-tieng-viet-GrLZDXr2Zk0>