Shape, square

Description automatically generated

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CNTT&TT**

-----🙞🙜🕮🙞🙜-----

**A picture containing text, sign

Description automatically generated**

**BÁO CÁO PROJECT 1**

***Giảng viên hướng dẫn: ThS. Vũ Đức Vượng***

***Sinh viên thực hiện : Bùi Đức Đăng***

***MSSV: 20200147***

**Đề tài: PHÁT HIỆN NGƯỜI TRẦM CẢM, NGUY CƠ TỰ TỬ QUA BÀI VIẾT TRÊN MẠNG XÃ HỘI BẰNG TRÍ TUỆ NHÂN TẠO**

Contents

[I. Giới thiệu chung 4](#_Toc146006337)

[II. Phân tích bài toán phát hiện trầm cảm tự tử sử dụng AI trên mạng xã hội 4](#_Toc146006338)

[1. Phân tích dữ liệu và tiền xử lý 4](#_Toc146006339)

[1.1 Trực quan hóa dữ liệu 4](#_Toc146006340)

[1.2 Tiền xử lý dữ liệu 5](#_Toc146006341)

[1.3 Phân tích độ dài văn bản sau tiền xử lý 6](#_Toc146006342)

[1.4 Biểu đồ từ phổ (Word Frequency) 7](#_Toc146006343)

[2. Chuẩn bị dữ liệu cho huấn luyện 8](#_Toc146006344)

[2.1 Chuyển đổi văn bản thành dãy số (Text to Sequence) 8](#_Toc146006345)

[2.2 Đối dãy số có chiều dài cố định (Padding Sequences) 8](#_Toc146006346)

[2.3 Xử lý nhãn lớp (Labels) 9](#_Toc146006347)

[3. Xây dựng mô hình AI 9](#_Toc146006348)

[3.1. Sử dụng các vectơ nhúng từ GloVe để xử lý văn bản. 9](#_Toc146006349)

[3.2.Xây dựng một mạng LSTM để huấn luyện mô hình. 10](#_Toc146006350)

[3.3. Huấn luyện mô hình 11](#_Toc146006351)

[3.4. Kết quả thu được 11](#_Toc146006352)

[4. Thu thập dữ liệu từ các dối tượng cần theo dõi trên facebook. 14](#_Toc146006353)

[4.1. Đăng nhập tự động 14](#_Toc146006354)

[4.2. Thu thập dữ liệu dạng văn bản trên các bài đăng tại trang cá nhân của từng đối tượng. 15](#_Toc146006355)

[5. Tiền xử lý dữ liệu trước khi đưa vào model để phân loại 18](#_Toc146006356)

[5.1. Giải quyết vấn đề model chỉ hiểu được tiếng Anh 18](#_Toc146006357)

[5.2.Xóa đi những ký hiệu đặc biệt, từ không mang nghĩa 20](#_Toc146006358)

[6. Dùng model đã học được để phân loại dữ liệu thu thập được 21](#_Toc146006359)

[6.1. Load model và tạo tokenizer 21](#_Toc146006360)

[6.2. Đọc dữ liệu từ data\_da\_chuyen\_doi , sử dụng hàm clean\_text để làm sạch văn bản, sau đó chuẩn hóa độ dài đầu vào cho model là 40 21](#_Toc146006361)

[6.3. Sử dụng model để phân loại sau đó gán nhãn cho dữ liệu trong file data\_da\_chuyen\_doi.csv 22](#_Toc146006362)

[7. Theo dõi và lấy thông tin bạn bè của những người có khả năng bị trầm cảm 22](#_Toc146006363)

[8.Phát đi cảnh báo 23](#_Toc146006364)

[9.Tổng kết 25](#_Toc146006365)

[Tài liệu tham khảo: 26](#_Toc146006366)

Mục lục ảnh

[Figure 1: Trực quan hóa dũ liệu 5](#_Toc146005964)

[Figure 2: Trực quan hóa dữ liệu 5](#_Toc146005965)

[Figure 3: Làm sạch dữ liệu 6](#_Toc146005966)

[Figure 4: Trực quan bằng đồ thị 7](#_Toc146005967)

[Figure 5: Biểu đồ cột tần suất xuất hiện các từ 8](#_Toc146005968)

[Figure 6: Chuyển văn bản thành dãy số bằng Tokenizer 8](#_Toc146005969)

[Figure 7: Chuyển đổi nhãn lớp 9](#_Toc146005970)

[Figure 8: Dùng GloVe để xử lý văn bản 10](#_Toc146005971)

[Figure 9: Xây dựng mô hình huấn luyện 10](#_Toc146005972)

[Figure 10: Huấn luyện mô hình 11](#_Toc146005973)

[Figure 11: Biểu đồ Accurancy 12](#_Toc146005974)

[Figure 12: Biểu đồ hàm Loss 13](#_Toc146005975)

[Figure 13: Kết quả dự đoán trên tập test 14](#_Toc146005976)

[Figure 14: Lưu lại mô hình đã huấn luyện được 14](#_Toc146005977)

[Figure 15: Login facebook 15](#_Toc146005978)

[Figure 16: Nội dung file links\_facebook.csv 16](#_Toc146005979)

[Figure 17: HÌnh ảnh bài đăng trên trang cá nhân 1 16](#_Toc146005980)

[Figure 18: HÌnh ảnh bài đăng trên trang cá nhân 2 17](#_Toc146005981)

[Figure 19: Dữ liệu lấy về được 17](#_Toc146005982)

[Figure 20: Chuyển tiếng Anh sang tiếng Việt dùng Translator 18](#_Toc146005983)

[Figure 21: Dữ liệu trước chuyển đổi 19](#_Toc146005984)

[Figure 22: Dữ liệu sau chuyển đổi 20](#_Toc146005985)

[Figure 23: Hàm clean\_text 20](#_Toc146005986)

[Figure 24:Load model và tạo tokenizer 21](#_Toc146005987)

[Figure 25:Làm sạch văn bản, chuẩn hóa độ dài đầu vào 21](#_Toc146005988)

[Figure 26: Phân loại và gán nhãn cho dữ liệu 22](#_Toc146005989)

[Figure 27: Lấy thông tin bạn bè 23](#_Toc146005990)

[Figure 28: Lấy link các đối tượng có nguy cơ đã lưu lại từ trước 24](#_Toc146005991)

[Figure 29: Lấy thông tin bạn bè ứng với từng đối tượng 24](#_Toc146005992)

[Figure 30: Phát đi cảnh báo 25](#_Toc146005993)

[Figure 31: Kết quả phát đi cảnh báo 25](#_Toc146005994)

# I. Giới thiệu chung

Trầm cảm là một trong những vấn đề tâm lý nghiêm trọng đang phát triển trên khắp thế giới, và việc phát hiện trầm cảm một cách sớm sẽ giúp cung cấp hỗ trợ và điều trị kịp thời cho những người bị ảnh hưởng. Trong báo cáo này, em sẽ phân tích bài toán phát hiện ngưỡng trầm cảm qua các bài viết trên mạng xã hội, với sự áp dụng của trí tuệ nhân tạo (AI).

Mạng xã hội đã trở thành nền tảng quan trọng cho việc thể hiện cảm xúc, chia sẻ suy nghĩ và tình trạng tâm trạng của mọi người. Điều này tạo ra một cơ hội để sử dụng trí tuệ nhân tạo (AI) để phát hiện trầm cảm thông qua việc phân tích dữ liệu từ các bài viết, bình luận, và thậm chí cả ảnh và video được chia sẻ trên các nền tảng mạng xã hội.

Ứng dụng AI trong việc phát hiện trầm cảm có thể có những lợi ích quan trọng như sau:

1. **Phát hiện sớm**: AI có khả năng theo dõi và phân tích dữ liệu liên tục, giúp phát hiện những biểu hiện sớm của trầm cảm và cung cấp sự hỗ trợ kịp thời.
2. **Đánh giá quy mô lớn**: AI có thể xử lý một lượng lớn dữ liệu từ các nguồn khác nhau trên mạng xã hội, giúp đánh giá tình hình trầm cảm ở mức độ quy mô lớn.
3. **Tích hợp dữ liệu đa dạng**: AI có khả năng tích hợp dữ liệu từ nhiều nguồn khác nhau, bao gồm văn bản, hình ảnh, âm thanh và video, để có cái nhìn toàn diện về tình trạng tâm trạng của người dùng. Tuy nhiên trong báo cáo này chỉ giới hạn trong việc sử dụng tới dữ liệu dạng văn bản.
4. **Bảo vệ quyền riêng tư**: AI cần được thiết lập cẩn thận để bảo vệ quyền riêng tư của người dùng và tuân thủ các quy định về bảo mật dữ liệu.

# II. Phân tích bài toán phát hiện trầm cảm tự tử sử dụng AI trên mạng xã hội

**Các thư viện sử dụng : numpy, pandas, sklearn, keras, tensorflow, seaborn, pickle, warnings** và một số thư viện khác.

## 1. Phân tích dữ liệu và tiền xử lý

Bộ dữ liệu làm việc : file **Suicide\_Detection.csv**

### 1.1 Trực quan hóa dữ liệu

**Bộ dữ liệu làm việc : file Suicide\_Detection.csv**

A screenshot of a computer

Description automatically generated

Figure 1: Trực quan hóa dũ liệu

Bộ dữ liệu gồm dữ liệu về nội dung các bài viết được thu thập trên mạng xã hội đã được dán nhãn suicide và non-suicide.

**Kiểm tra phân phối lớp của dữ liệu**

A screenshot of a computer code

Description automatically generated

Figure 2: Trực quan hóa dữ liệu

Điều này cho thấy dữ liệu được phân chia đều giữa hai lớp 'suicide' và 'non-suicide' với cùng số lượng mẫu, mỗi lớp có 116,037 mẫu.

### 1.2 Tiền xử lý dữ liệu

Chia dữ liệu thành tập test và train với tỉ lệ 2:8.

Sau đó, tiến hành quá trình tiền xử lý dữ liệu với các bước sau:

* Biến đổi văn bản thành chữ thường.
* Gỡ bỏ các ký tự đặc biệt như dấu câu và ký tự không phải chữ cái hoặc số.
* Gỡ bỏ các từ không mang nghĩa như "and", "is", để tập trung vào những từ quan trọng hơn trong văn bản.

A computer screen shot of a computer code

Description automatically generated

Figure 3: Làm sạch dữ liệu

Tạo hàm clean\_text để thực hiện các bước trên.

### 1.3 Phân tích độ dài văn bản sau tiền xử lý

Vẽ đồ thị thể hiện sự phân bổ của độ dài của văn bản sau khi tiền xử lý

A graph with numbers and lines

Description automatically generated

Figure 4: Trực quan bằng đồ thị

Từ dồ thị trên ta thấy sự phân bổ không đồng đều có đoạn ngắn đoạn dài. Vì vậy tại các bước sau ta phải chuẩn hóa độ dài đầu vào của văn bản.

### 1.4 Biểu đồ từ phổ (Word Frequency)

Tạo biểu đồ cột về tần suất xuất hiện của các từ trong dữ liệu sau tiền xử lý:

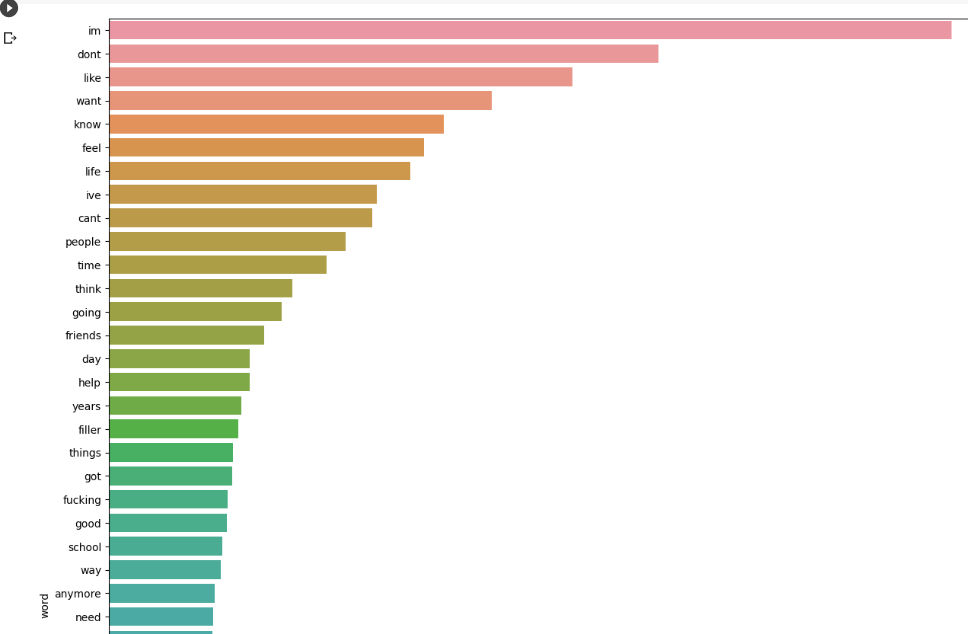


Figure 5: Biểu đồ cột tần suất xuất hiện các từ

## 2. Chuẩn bị dữ liệu cho huấn luyện

### 2.1 Chuyển đổi văn bản thành dãy số (Text to Sequence)

Sau khi đã thực hiện tiền xử lý dữ liệu văn bản, chúng ta cần chuyển đổi các văn bản thành các dãy số để có thể đưa vào mô hình học máy.

**A screenshot of a computer

Description automatically generated**

Figure 6: Chuyển văn bản thành dãy số bằng Tokenizer

Sử dụng phương thức: **texts\_to\_sequences** trongthư viện: **Tokenizer** ( Keras) để chuyển văn bản thành dãy số.

### 2.2 Đối dãy số có chiều dài cố định (Padding Sequences)

Để đảm bảo rằng tất cả các dãy số có cùng chiều dài, chúng ta sử dụng quy trình gọi là "padding". Điều này giúp đảm bảo rằng tất cả các mẫu đều có cùng độ dài, đồng thời cung cấp cho mô hình đầu vào có kích thước đồng nhất. Đoạn mã sau thực hiện padding cho dãy số:

A screenshot of a computer code

Description automatically generated

Sử dụng phương thức:  **pad\_sequences** trongthư viện **pad\_sequences** ( Keras) để cố định chiều dài văn bản là 40.

### 2.3 Xử lý nhãn lớp (Labels)

Để huấn luyện mô hình, chúng ta cần chuyển đổi nhãn lớp từ dạng chuỗi sang dạng số nguyên. Để làm điều này, chúng ta sử dụng **LabelEncoder**. Đoạn mã sau thực hiện chuyển đổi nhãn lớp:

A screenshot of a computer

Description automatically generated

Figure 7: Chuyển đổi nhãn lớp

Sử dụng phương thức: **fit\_transform**, **transform** trongthư viện **LabelEncoder** ( scikit-learn) để chuyển đổi nhãn lớp từ dạng chuỗi sang dạng số nguyên. **fit\_transform** được sử dụng trên tập huấn luyện để tạo bản đồ ánh xạ từ chuỗi nhãn sang số nguyên. Sau đó, **transform** được sử dụng trên tập kiểm tra để chuyển đổi dựa trên cùng một bản đồ ánh xạ như trong tập huấn luyện.

## 3. Xây dựng mô hình AI

### 3.1. Sử dụng các vectơ nhúng từ GloVe để xử lý văn bản.

Sử dụng các vectơ nhúng từ GloVe đã được huấn luyện trên một lượng lớn văn bản (840 tỷ từ và 300 chiều) để xử lý văn bản.

A screenshot of a computer program

Description automatically generated

Figure 8: Dùng GloVe để xử lý văn bản

Một vòng lặp được sử dụng để duyệt qua từng từ trong từ điển **tokenizer.word\_index**. Với mỗi từ sẽ kiểm tra xem có một vectơ nhúng tương ứng trong **glove\_embedding** không. Nếu có, vectơ nhúng này được gán vào hàng tương ứng của **embedding\_matrix**. Điều này giúp tạo ra ma trận nhúng cho toàn bộ từ điển của tập dữ liệu.

### 3.2.Xây dựng một mạng LSTM để huấn luyện mô hình.

A computer code with text

Description automatically generated with medium confidence

Figure 9: Xây dựng mô hình huấn luyện

1. Xây dựng một model tuần tự với đầu vào kích thước là 40.
2. Tiếp theo thêm một lớp nhúng để biểu diễn các từ dưới dạng vectơ nhúng **trainable=False** để không cập nhật các giá trị nhúng trong quá trình đào tạo.
3. Lớp **LSTM** với 20 đơn vị đầu ra và **return\_sequences=True** cho biết rằng nó sẽ trả về dãy đầu ra tại mỗi thời điểm thay vì chỉ ở thời điểm cuối cùng.
4. Thêm lớp **Global Max Pooling 1D** để lấy giá trị lớn nhất từ mỗi dãy đầu ra của LSTM, giúp giảm kích thước dữ liệu đầu ra xuống một chiều.
5. Lớp **Dropout** được sử dụng để ngẫu nhiên loại bỏ một phần các đơn vị đầu ra trong quá trình đào tạo, với tỷ lệ 30% ở đây. Điều này giúp tránh tình trạng overfitting.
6. Lớp **Dense** được thêm vào tiếp với 256 đơn vị đầu ra và sử dụng hàm kích hoạt **ReLU**.
7. Thêm lớp **Dense** output lớp này có một đơn vị đầu ra và sử dụng hàm kích hoạt sigmoid. Đây là lớp đầu ra của mô hình, dùng để dự đoán một giá trị nhị phân (0 hoặc 1) cho bài toán phân loại.
8. Mô hình trên sẽ sử dụng các tham số **optimizer**: Sử dụng tối ưu hóa SGD (Stochastic Gradient Descent) với learning rate là 0.1 và momentum là 0.09. **loss**: sử dụng hàm mất mát là **'binary\_crossentropy'** cho bài toán phân loại nhị phân. Vá metrics: Đánh giá mô hình bằng chỉ số **'accuracy'**

### 3.3. Huấn luyện mô hình

**A black and white text

Description automatically generated**

Figure 10: Huấn luyện mô hình

### 3.4. Kết quả thu được

#### 3.4.1. Biểu đồ Accuracy

**A graph with blue and orange lines

Description automatically generated**

Figure 11: Biểu đồ Accurancy

Từ đồ thị ta thấy mô hình hội tụ khá tốt. Độ chính xác đều cao trên cả tập train và tập val.

#### 3.4.2. Biểu đồ hàm Loss

A graph with blue lines and numbers

Description automatically generated

Figure 12: Biểu đồ hàm Loss

Từ đồ thị ta thấy mô hình hội tụ khá tốt. Giá trị Loss giảm tốt trên cả hai tập train và val và đủ nhỏ .

#### 3.4.3. Kết quả dự đoán trên tập test

**A screenshot of a computer

Description automatically generated**

Figure 13: Kết quả dự đoán trên tập test

Các giá trị preision, recall , f1-score đều rất cao chứng tỏ mô hình đang hoạt động tốt.

#### 3.4.4. Lưu lại mô hình đã huấn luyện được

A close-up of a computer screen

Description automatically generated

Figure 14: Lưu lại mô hình đã huấn luyện được

Mô hình đã huấn luyện được lưu trong file **Suicidal\_0.h5.**

## 4. Thu thập dữ liệu từ các dối tượng cần theo dõi trên facebook.

Sử dụng Selenium để tự động đăng nhập vào tài khoản Facebook và thu thập dữ liệu từ các đối tượng cần theo dõi trên Facebook.

### 4.1. Đăng nhập tự động

Sử dụng thư viện **chromedriver** để dùng trình duyệt **Chrome** .

A screenshot of a computer

Description automatically generated

Figure 15: Login facebook

### 4.2. Thu thập dữ liệu dạng văn bản trên các bài đăng tại trang cá nhân của từng đối tượng.

#### 4.2.1. Thu thập link đường dẫn tới trang cá nhân của từng đối tượng

Thu thập tên, link trang cá nhân trên face lưu tại file **links\_facebook.csv**. File có 3 cột tương ứng **ID, link, Name**

Như hình :

A screenshot of a computer

Description automatically generated

Figure 16: Nội dung file links\_facebook.csv

#### 4.2.2. Thu thập dữ liệu những bài viết mới nhất trên trang cá nhân từng đối tượng đã lưu trong file links\_facebook.csv.

Sử dụng selenium để load trang, click các ô cần thiết, sau đó tìm các thẻ HTML chứa dữ liệu cần và lấy về.

**Bài đăng trên trang cá nhân:  
A screenshot of a computer

Description automatically generated**

Figure 17: HÌnh ảnh bài đăng trên trang cá nhân 1

**A screenshot of a chat

Description automatically generated**

Figure 18: HÌnh ảnh bài đăng trên trang cá nhân 2

**Dữ liệu lấy về được:  
A screenshot of a computer

Description automatically generated**

Figure 19: Dữ liệu lấy về được

Tương tự như vậy cho các cá nhân khác và lưu lại.

Dữ liệu đã lấy về được lưu lại trong file **output\_data.csv** gồm 3 cột **link\_facebook, Content , Date**.

Trong đó Content là nới chưa dữ liệu nội dung bài viết đã lấy được.

## 5. Tiền xử lý dữ liệu trước khi đưa vào model để phân loại

### 5.1. Giải quyết vấn đề model chỉ hiểu được tiếng Anh

**Đề suât giải pháp:** chuyển đổi nội dung các bài viết qua tiếng Anh sau đó mới dùng model để phân loại.

**Giải quyết** :

* Dùng google dịch : quá phức tạp khi phải đẩy từng Content lên web sau đó mới lấy về . Quá phức tạp và tốn thời gian khi phải dừng chương trình để chờ load web.
* Sử dụng thư viện **Translator** một thư viện mã nguồn mở hỗ trợ việc chuyển đồi này bằng các model đã được huấn luyện từ trước. Hiệu suất cao tiện lợi trên máy tính cá nhân.

Thực hiện chuyển đổi tiếng Việt qua tiếng Anh bằng **Translator** :

**A screen shot of a computer program

Description automatically generated**

Figure 20: Chuyển tiếng Anh sang tiếng Việt dùng Translator

Ta sẽ đọc từng thành phần trong file **out\_put\_data.csv** sau đó thực hiện chuyển đồi cột **Content** sang tiếng Anh và lưu chúng lại trong file **data\_da\_chuyen\_doi.csv.**

**Dữ liệu trước và sau khi chuyển đổi :**

**A screenshot of a computer

Description automatically generated**

Figure 21: Dữ liệu trước chuyển đổi

A screenshot of a computer

Description automatically generated

Figure 22: Dữ liệu sau chuyển đổi

### 5.2.Xóa đi những ký hiệu đặc biệt, từ không mang nghĩa

Sử dụng hàm **clean\_text.**

**A screen shot of a computer code

Description automatically generated**

Figure 23: Hàm clean\_text

## 6. Dùng model đã học được để phân loại dữ liệu thu thập được

### 6.1. Load model và tạo tokenizer

A computer screen with colorful text

Description automatically generated

Figure 24:Load model và tạo tokenizer

### 6.2. Đọc dữ liệu từ data\_da\_chuyen\_doi , sử dụng hàm clean\_text để làm sạch văn bản, sau đó chuẩn hóa độ dài đầu vào cho model là 40

A computer screen shot of a program code

Description automatically generated

Figure 25:Làm sạch văn bản, chuẩn hóa độ dài đầu vào

### 6.3. Sử dụng model để phân loại sau đó gán nhãn cho dữ liệu trong file data\_da\_chuyen\_doi.csv

A computer screen shot of text

Description automatically generated

Figure 26: Phân loại và gán nhãn cho dữ liệu

## 7. Theo dõi và lấy thông tin bạn bè của những người có khả năng bị trầm cảm

Duyệt dữ liệu đã gán nhãn trong data\_da\_chuyen\_doi.csv để lấy ra các dối tượng và số lượng bài viết có khả năng trầm cảm .

Từ các đối tượng đã lọc ra được tiến hành lấy link facebook của bạn bè bằng cách sử dụng **selenium** tương tự như khi lấy dữ liệu bài viết.

Ta sẽ kiểm tra xem đối tượng đã có thông tin bạn bè chưa trước khi thực hiện để tránh lấy trùng lặp. Nếu chưa có ta sẽ tiến hành lấy tối đa 5 đường link của bạn bè, tất cả lưu lại vào file **doi\_tuong\_can\_theo\_doi.csv.**

**A screen shot of a computer program

Description automatically generated**

Figure 27: Lấy thông tin bạn bè

Từ đây có thể tiến hành theo dõi kỹ hơn một số đối tượng có nguy cơ cao hơn. Có thể nghiên cứu tính toán trọng số dựa vào ngưỡng phân loại **threshold** của model và **tần suất xuất hiện** cũng như **số lượng** của các bài viết có khả năng trầm cảm tự tử từ đó làm cơ sở để phát đi cảnh báo.

## 8.Phát đi cảnh báo

Khi đối tượng được theo dõi đạt tới một nguy cơ nhất định sẽ tiến hành phát đi cảnh báo, nội dung cảnh báo cũng sẽ tùy theo mức độ. Cơ chế tính trọng số và ngưỡng phát đi cảnh báo vẫn trong quá trính nghiên cứu. Tuy nhiên ta sẽ làm chức năng phát đi cảnh báo luôn để khi hoàn thành cơ chế tính trọng số sẽ có thể phát đi được ngay.

Công cụ sử dụng : thư viện **pyautogui** , **pyperclip.**

Hai thư viện trên giúp copy văn bản có sẵn và tự động gửi đi.

Đọc file **doi\_tuong\_can\_theo\_doi.csv** để lấy link của các đối tượng

A computer screen with many colorful text

Description automatically generated

Figure 28: Lấy link các đối tượng có nguy cơ đã lưu lại từ trước

Với mỗi một đối tượng sẽ lấy thông tin bạn bè đã có từ trước lưu vào **friend\_links**

A computer screen shot of a code

Description automatically generated

Figure 29: Lấy thông tin bạn bè ứng với từng đối tượng

Khi đã có link của bạn bè , sử dụng **selenium** để mở trình duyệt ,đăng nhập, truy cập tới

trang cá nhân của bạn bè , mở **message** sau đó gửi đi thông điệp cảnh báo.

A computer screen with text on it

Description automatically generated

Figure 30: Phát đi cảnh báo

Kết quả tự động gửi đi như hình

A screenshot of a chat

Description automatically generated

Figure : Kết quả phát đi cảnh báo

Các nôi dung thông điệp gửi đi cần phát triển thêm.

## 9.Tổng kết

Mô hình xử lý tiếng Việt hiệu suất chưa cao. Các chức năng phân loại theo dõi , phát đi cảnh báo chưa thực sự hoàn thiện cần phải tôi ưu và phát triển thêm. Cơ chế tính trọng số và tính ngưỡng phát cảnh báo cần nghiên cứu thêm và thực hiện.

Định hướng phát triển:

* Cần nghiên cứu phát triển một mô hình có khả năng hiểu tiếng Việt tốt để áp dụng.
* Nghiên cứu cơ chế tính trọng số và ngưỡng phát cảnh sau đó triển khai và đánh giá.
* Mở rộng thêm sang các mạng xã hội khác ngoài facebook.
* Nghiên cứu phát triển một chatbot chuyên về lĩnh vực này để có thể hỗ trợ cho những người bị trầm cảm.

# Tài liệu tham khảo:

<https://numpy.org/doc/stable/>

<https://pandas.pydata.org/docs/>

<https://keras.io/guides/sequential_model/>

<https://www.tensorflow.org/guide>

<https://seaborn.pydata.org/tutorial.html>

<https://docs.python.org/3/library/pickle.html>

<https://docs.python.org/3/library/warnings.html>

<https://www.kaggle.com/datasets/szamil/who-suicide-statistics>

<https://www.kaggle.com/datasets/russellyates88/suicide-rates-overview-1985-to-2016>

<https://www.kaggle.com/code/abhijitsingh001/suicidal-thought-detection>