BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM
Khoa: Công nghệ thông tin

BÁO CÁO
ĐỒ ÁN CƠ SỞ

"GOM NHÓM NỘI DUNG PHẢN HỒI TÍCH CỰC, TIÊU CỰC THEO CHỦ ĐỀ TRÊN MẠNG XÃ HỘI"

Giảng viên hướng dẫn: Thạc sỹ Nguyễn Đình Ánh

Sinh viên thực hiện:		
Trịnh Thị Thanh Nhung	1611061638	16DTHB5
Phạm Thị Thanh Mai	1611061594	16DTHB5
Đặng Thị Bảo Nghi	1611062130	16DTHB5

Thành phố Hồ Chí Minh, 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM
Khoa: Công nghệ thông tin

BÁO CÁO

ĐỒ ÁN CƠ SỞ

"GOM NHÓM NỘI DUNG PHẢN HỒI TÍCH CỰC, TIÊU CỰC THEO CHỦ ĐỀ TRÊN MẠNG XÃ HỘI"

Giảng viên hướng dẫn: Thạc sỹ Nguyễn Đình Ánh

Sinh viên thực hiện:	Trịnh Thị Thanh Nhung	1611061638	16DTHB5
	Phạm Thị Thanh Mai	1611061594	16DTHB5
	Đặng Thị Bảo Nghi	1611062130	16DTHB5

Thành phố Hồ Chí Minh, 2019

LỜI NÓI ĐẦU

Ngày nay ngành công nghệ thông tin là một ngành khoa học đang trên đà phát triển mạnh và ứng dụng rộng rãi trên nhiều lĩnh vực. Cùng với xu hướng phát triển của các phương tiện truyền thông thì việc sử dụng Internet ngày càng phổ biến. Truy cập Internet, giúp chúng ta có được một kho thông tin khổng lồ phục vụ mọi nhu cầu, mục đích của chúng ta chỉ với vài thao tác đơn giản.

Việc sử dụng mạng xã hội hiện nay dường như là hoạt động không thể thiếu trong ngày của mỗi người. Ngoài việc nắm bắt được các thông tin ra, người sử dụng có thể thỏa sức để lại đánh giá, bình luận để biểu đạt ý kiến của bản thân. Mỗi người đều có suy nghĩ riêng của mình, cho nên ý kiến của mỗi người cũng khác nhau. Có người tán thành, khen ngợi một việc gì đó nhưng có người lại có suy nghĩ ngược lại. Điều này tạo nên hai luồng phản hồi trái ngược trên mạng xã hội: tích cực và tiêu cực.

Nhận thức được điều đó, chúng em đã vận dụng các kiến thức được học trên lớp cùng với các thông tin tìm hiểu được để xây dựng nên đồ án này. Cho phép gom nhóm các phản hồi tích cực, tiêu cực theo chủ đề trên mạng xã hội. Trong quá trình làm còn nhiều sai sót, chúng em mong nhận được những đánh giá và góp ý từ quý thầy cô.

Sinh viên thực hiện

Trịnh Thị Thanh Nhung
Phạm Thị Thanh Mai
Đặng Thị Bảo Nghi

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn quý thầy cô đã giúp đỡ chúng em thực hiện đề tài này. Đặc biệt thầy Nguyễn Đình Ánh đã tận tình hướng dẫn, giúp đỡ, chỉ bảo và đôn đốc chúng em trong suốt quá trình thực hiện đồ án.

Đồng thời chúng em cũng xin trân trọng cảm ơn sâu sắc đến tất cả các quý thầy cô trường Đại Học Công Nghệ TP.HCM – HUTECH, các quý thầy cô khoa Công Nghệ Thông Tin đã truyền đạt những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đồ án này.

Tuy nhiên, do thời gian có hạn nên chúng em không thể phát huy hết ý tưởng, khả năng hỗ trợ của ngôn ngữ và kỹ thuật lập trình vào đề tài gom nhóm các phản hồi tích cực, tiêu cực này.

Mặc dù đã cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý thầy cô và các bạn.

LỜI CAM ĐOAN

Chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em. Các kết quả, số liệu nêu trong báo cáo là trung thực và chưa được công bố trong các bài báo cáo khác. Nếu không đúng như đã nêu trên, chúng em xin hoàn toàn chịu trách nhiệm về đề tài của mình.

Sinh viên cam kết

Trịnh Thị Thanh Nhung

Phạm Thị Thanh Mai

Đặng Thị Bảo Nghi

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH

Hình 4.10. Sheet Positive lưu các tweet mang tính tích cực.

Hình 4.11. Sheet Negative lưu các tweet mang tính tiêu cực.

Chương 1: TỔNG QUAN

1.1. Hiện trạng:

Mỗi thời đại sẽ có những cách khác nhau để liên lạc, trao đổi thông tin. Ngày xưa, con người thường viết thư và chờ đợi những bức thư phản hồi, thời gian rất rất là lâu vì khoảng cách xa xôi, vì phương tiện vận chuyển. Nhưng ngày nay với cuộc cách mạng khoa học công nghệ 4.0 thì những bức thư đó được thay thế bằng những cú click, những dòng enter của các trang mạng xã hội.

Mạng xã hội đã kết nối con người khắp nơi trên thế giới, xóa nhòa khoảng cách về không gian, thời gian nhờ tốc độ nhanh chóng và sự tiện lợi của nó. Nhưng cũng vì quá lạm dụng nó mà nhiều người hiện nay tự tập cho mình một lối sống không lành mạnh.

Trên các trang mạng xã hội như Facebook, Instagram, Zalo, Twitter,... và vô số trang mạng xã hội khác nữa, việc giao tiếp trở nên quá dễ dàng, khoảng cách như được thu hẹp lại, vì thế khiến cho mỗi người đều đam mê, yêu thích. Nhiều bạn trẻ hiện nay cho rằng trang cá nhân trên mạng xã hội là nơi để họ thoải mái thể hiện quan điểm mà không cần suy nghĩ về hậu quả của những gì mình viết ra. Đừng nghĩ rằng ta có thể ngồi sau bàn phím máy tính và thoải mái bộc lộ suy nghĩ của mình cũng như lời nói, vì ngôn từ viết ra có sức "sát thương" rất cao.

Có thể trong một bài viết, có người bộc lộ sự ưa thích và giành những lời khen ngợi cho bài viết đó. Nhưng có những người lại tỏ thái độ không thích, từ đó có những lời lẽ bình luận không "hoa mĩ" về bài viết hay người đăng tải. Và thế là xuất hiện hai luồng phản hồi trái chiều nhau: tích cực và tiêu cực. Tình trạng này, trên mạng xã hội hiện nay không phải là hiếm hoi nữa.

Báo cáo này nghiên cứu về trang mạng xã hội Twitter, cho nên chủ yếu sẽ chỉ tập trung vào Twitter.

Twitter là một dịch vụ mạng xã hội trực tuyến miễn phí cho phép người sử dụng đọc, nhắn và cập nhật các mẩu tin nhỏ gọi là tweets, một dạng tiểu blog.

Hiện nay Twitter đã hỗ trợ người dùng đăng các Tweet dưới dạng đoạn hội thoại, đăng ảnh, video, ảnh động, và tính năng cập nhật "Khoảnh khắc".

Những mẩu tweet được giới hạn tối đa 280 ký tự được lan truyền nhanh chóng trong phạm vi nhóm bạn của người nhắn hoặc có thể được trưng rộng rãi cho mọi người. Việc giới hạn ký tự này mang đến cho cộng đồng mạng một hình thức tốc ký đáng chú ý và được sử dụng rộng rãi như SMS – tin nhắn thông thường.

1.2. Nhiệm vụ đồ án:

1.2.1. Mục đích:

Đồ án "Gom nhóm các phản hồi tích cực, tiêu cực theo chủ đề trên mạng xã hội" được thực hiện dựa trên bốn mục đích chính:

- ✓ Phân loại các phản hồi ra để dễ dàng hơn trong công tác quản lý.
- ✓ Quản lý kịp thời các thông tin phản hồi hoặc bình luận tiêu cực giúp ngăn chặn nó trở thành thảm họa.
- ✓ Xây dựng thương hiệu luôn là mục tiêu hàng đầu và cuối cùng mà các doanh nghiệp hướng tới, từ đó các phản hồi càng cần được quản lý chặt chẽ hơn.
- ✓ Xây dựng bảng đánh giá mức độ yêu thích đối với một sự vật, sự việc nào đó. Thuận lợi cho chiến lược Marketing của các thương hiệu kinh doanh.

1.2.2. Yêu cầu:

1.2.2.1. <u>Về lý thuyết:</u>

- ✓ Đảm bảo có thể lấy được tất cả các bình luận, phản hồi của mọi người về các vấn đề có liên quan.
- ✓ Phân loại được các phản hồi trên thành tích cực và tiêu cực, từ đó có những xử lý kịp thời trong công các quản lý.

1.2.2.2. <u>Nhiệm vụ đề ra:</u>

- ✓ Tìm hiểu về ngôn ngữ lập trình Python.

- ✓ Nghiên cứu cách truy xuất bình luận của mọi người trên mạng xã hội (cụ thể là trên mạng xã hội Twitter).
- ✓ Tiến hành phân tích dữ liệu đã lấy được thành hai luồng: tích cực và tiêu cực.
- ✓ Trình bày kết quả dưới dạng file Excel.

1.3. Cấu trúc đồ án:

Chương 1: TỔNG QUAN: Giới thiệu về hiện trạng của việc sử dụng mạng xã hội hiện nay. Trình bày cấu trúc đồ án, nhiệm vụ của đồ án: bao gồm mục đích và yêu cầu.

Chương 2: CƠ SỞ LÝ THUYẾT: Giới thiệu ngôn ngữ, công nghệ, môi trường sử dụng để phát triển đề tài.

Chương 3: PHÂN TÍCH VÀ THIẾT KẾ: Trình bày các bước phân tích và thiết kế nên sản phẩm.

Chương 4: KẾT QUẢ THỰC NGHIỆM: Cung cấp một số kết quả thực hiện được, bao gồm cả những kết quả bị lỗi và chỉ ra lý do tại sao.

Chương 5: TỔNG KẾT: Tóm tắt kết quả đã thực hiện, đánh giá ưu – khuyết điểm và trình bày hướng phát triển trong tương lai.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ lập trình Python:

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã "bành trướng" sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

2.1.1. Các bản hiện thực:

Python được viết từ những ngôn ngữ khác, tạo ra những bản hiện thực khác nhau. Bản hiện thực Python chính, còn gọi là **CPython**, được viết bằng C, và được phân phối kèm một thư viện chuẩn lớn được viết hỗn hợp bằng C và Python. CPython có thể chạy trên nhiều nền và khả chuyển trên nhiều nền khác. Dưới đây là các nền, trên đó CPython có thể chạy:

- ✓ Các hệ điều hành họ Unix: AIX, Darwin, FreeBSD, Mac OS X, NetBSD, Linux, OpenBSD, Solaris,...
- ✓ Các hệ điều hành dành cho máy desktop: Amiga, AROS, BeOS, Mac OS 9, Microsoft Windows, OS/2, RISC OS.

- ✓ Các hệ thống nhúng và các hệ đặc biệt: GP2X, Máy ảo Java, Nokia 770 Internet Tablet, Palm OS, PlayStation 2, PlayStation Portable, Psion, QNX, Sharp Zaurus, Symbian OS, Windows CE/Pocket PC, Xbox/XBMC, VxWorks.
- ✓ Các hệ máy tính lớn và các hệ khác: AS/400, OS/390, Plan 9 from Bell Labs, VMS, z/OS.

Ngoài CPython, còn có hai hiện thực Python khác: Jython cho môi trường Java và IronPython cho môi trường .NET và Mono.

2.1.2. Khả năng mở rộng:

Python có thể được mở rộng: nếu ta biết sử dụng C, ta có thể dễ dàng viết và tích hợp vào Python nhiều hàm tùy theo nhu cầu. Các hàm này sẽ trở thành hàm xây dựng sẵn (*built-in*) của Python. Ta cũng có thể mở rộng chức năng của trình thông dịch, hoặc liên kết các chương trình Python với các thư viện chỉ ở dạng nhị phân (như các thư viện đồ họa do nhà sản xuất thiết bị cung cấp). Hơn thế nữa, ta cũng có thể liên kết trình thông dịch của Python với các ứng dụng viết từ C và sử dụng nó như là một mở rộng hoặc một ngôn ngữ dòng lệnh phụ trợ cho ứng dụng đó.

2.1.3. Trình thông dịch:

Python là một ngôn ngữ lập trình dạng thông dịch, do đó có ưu điểm tiết kiệm thời gian phát triển ứng dụng vì không cần phải thực hiện biên dịch và liên kết. Trình thông dịch có thể được sử dụng để chạy file script, hoặc cũng có thể được sử dụng theo cách tương tác. Ở chế độ tương tác, trình thông dịch Python tương tự shell của các hệ điều hành họ Unix, tại đó, ta có thể nhập vào từng biểu thức rồi gõ `Enter`, và kết quả thực thi sẽ được hiển thị ngay lập tức. Đặc điểm này rất hữu ích cho người mới học, giúp họ nghiên cứu tính năng của ngôn ngữ; hoặc để các lập trình viên chạy thử mã lệnh trong suốt quá trình phát triển phần mềm. Ngoài ra, cũng có thể tận dụng đặc điểm này để thực hiện các phép tính như với máy tính bỏ túi.

2.1.4. Lệnh và cấu trúc điều khiển:

Mỗi câu lệnh trong Python nằm trên một dòng mã nguồn. Ta không cần phải kết thúc câu lệnh bằng bất kì ký tự gì. Cũng như các ngôn ngữ khác, Python cũng có các cấu trúc điều khiển. Chúng bao gồm:

- ✓ Cấu trúc rẽ nhánh: cấu trúc `if` (có thể sử dụng thêm `elif` hoặc `else`), dùng để thực thi có điều kiện một khối mã cụ thể.
- ✓ Cấu trúc lặp, bao gồm:
  - Lệnh `while`: chạy một khối mã cụ thể cho đến khi điều kiện lặp có giá trị `false`.
  - Vòng lặp `for`: lặp qua từng phần tử của một dãy, mỗi phần tử sẽ được đưa vào biến cục bộ để sử dụng với khối mã trong vòng lặp.
- ✓ Python cũng có từ khóa `class` dùng để khai báo lớp (sử dụng trong lập trình hướng đối tượng) và lệnh `def` dùng để định nghĩa hàm.

2.1.5. Hệ thống kiểu dữ liệu:

Python sử dụng hệ thống kiểu *duck typing*, còn gọi là *latent typing* (tự động xác định kiểu). Có nghĩa là, Python không kiểm tra các ràng buộc về kiểu dữ liệu tại thời điểm dịch, mà là tại thời điểm thực thi. Khi thực thi, nếu một thao tác trên một đối tượng bị thất bại, thì có nghĩa là đối tượng đó không sử dụng một kiểu thích hợp.

Python cũng là một ngôn ngữ định kiểu mạnh. Nó cấm mọi thao tác không hợp lệ, ví dụ cộng một con số vào chuỗi ký tự.

Sử dụng Python, ta không cần phải khai báo biến. Biến được xem là đã khai báo nếu nó được gán một giá trị lần đầu tiên. Căn cứ vào mỗi lần gán, Python sẽ tự động xác định kiểu dữ liệu của biến. Python có một số kiểu dữ liệu thông dụng sau:

- ✓ `int`, `long`: số nguyên (trong phiên bản 3.x long được nhập vào trong kiểu int). Độ dài của kiểu số nguyên là tùy ý, chỉ bị giới hạn bởi bộ nhớ máy tính.
- ✓ `float`: số thực.
- ✓ `complex`: số phức, chẳng hạn `5+4j`.
- ✓ `list`: dãy trong đó các phần tử của nó có thể được thay đổi, chẳng hạn `[8, 2, 'b', -1.5]`. Kiểu dãy khác với kiểu mảng (*array*) thường gặp trong các ngôn ngữ lập trình ở chỗ các phần tử của dãy không nhất thiết có kiểu giống nhau. Ngoài ra phần tử của dãy còn có thể là một dãy khác.
- ✓ `tuple`: dãy trong đó các phần tử của nó không thể thay đổi.
- ✓ `str`: chuỗi ký tự. Từng ký tự trong chuỗi không thể thay đổi. Chuỗi ký tự được đặt trong dấu nháy đơn, hoặc nháy kép.
- ✓ `dict`: từ điển, còn gọi là "hashtable": là một cặp các dữ liệu được gắn theo kiểu {từ khóa: giá trị}, trong đó các từ khóa trong một từ điển nhất thiết phải khác nhau. Chẳng hạn `{1: "Python", 2: "Pascal"}`.
- ✓ `set`: một tập không xếp theo thứ tự, ở đó, mỗi phần tử chỉ xuất hiện một lần.

Ngoài ra, Python còn có nhiều kiểu dữ liệu khác.

2.1.6. Module:

Python cho phép chia chương trình thành các module để có thể sử dụng lại trong các chương trình khác. Nó cũng cung cấp sẵn một tập hợp các modules chuẩn mà lập trình viên có thể sử dụng lại trong chương trình của họ. Các module này cung cấp nhiều chức năng hữu ích, như các hàm truy xuất tập tin, các lời gọi hệ thống, trợ giúp lập trình mạng (socket),…

2.1.7. Đa năng:

Python là một ngôn ngữ lập trình đơn giản nhưng rất hiệu quả.

- ✓ So với Unix shell, Python hỗ trợ các chương trình lớn hơn và cung cấp nhiều cấu trúc hơn.
- ✓ So với C, Python cung cấp nhiều cơ chế kiểm tra lỗi hơn. Nó cũng có sẵn nhiều kiểu dữ liệu cấp cao, ví dụ như các mảng (*array*) linh hoạt và từ điển (*dictionary*) mà ta sẽ phải mất nhiều thời gian nếu viết bằng C.

Python là một ngôn ngữ lập trình cấp cao có thể đáp ứng phần lớn yêu cầu của lập trình viên:

- ✓ Python thích hợp với các chương trình lớn hơn cả AWK và Perl.
- ✓ Python được sử dụng để lập trình Web. Nó có thể được sử dụng như một ngôn ngữ kịch bản.
- ✓ Python được thiết kế để có thể nhúng và phục vụ như một ngôn ngữ kịch bản để tuỳ biến và mở rộng các ứng dụng lớn hơn.
- ✓ Python được tích hợp sẵn nhiều công cụ và có một thư viện chuẩn phong phú, Python cho phép người dùng dễ dàng tạo ra các dịch vụ Web, sử dụng các thành phần COM hay CORBA, hỗ trợ các loại định dạng dữ liệu Internet như email, HTML, XML và các ngôn ngữ đánh dấu khác. Python cũng được cung cấp các thư viện xử lý các giao thức Internet thông dụng như HTTP, FTP,...
- ✓ Python có khả năng giao tiếp đến hầu hết các loại cơ sở dữ liệu, có khả năng xử lý văn bản, tài liệu hiệu quả, và có thể làm việc tốt với các công nghệ Web khác.
- ✓ Python đặc biệt hiệu quả trong lập trình tính toán khoa học nhờ các công cụ Python Imaging Library, pyVTK, MayaVi 3D Visualization Toolkits, Numeric Python, ScientificPython,...
- ✓ Python có thể được sử dụng để phát triển các ứng dụng desktop. Lập trình viên có thể dùng wxPython, PyQt, PyGtk để phát triển các ứng dụng giao diện đồ họa (GUI) chất lượng cao. Python còn hỗ trợ các nền

tảng phát triển phần mềm khác như MFC, Carbon, Delphi, X11, Motif, Tk, Fox, FLTK,...

- ✓ Python cũng có sẵn một *unit testing framework* để tạo ra các bộ test (*test suites*).

2.1.8. Multiple paradigms (đa biến hóa):

Python là một ngôn ngữ đa biến hóa (multiple paradigms). Có nghĩa là, thay vì ép buộc mọi người phải sử dụng duy nhất một phương pháp lập trình, Python lại cho phép sử dụng nhiều phương pháp lập trình khác nhau: hướng đối tượng, có cấu trúc, chức năng, hoặc chỉ hướng đến một khía cạnh. Python kiểu động và sử dụng bộ thu gom rác để quản lý bộ nhớ. Một đặc điểm quan trọng nữa của Python là giải pháp tên động, kết nối tên biến và tên phương thức lại với nhau trong suốt thực thi của chương trình.

2.2. TextBlob:

TextBlob đứng trên đôi vai khổng lồ của NLTK và *pattern.en*. Nhờ thế TextBlob cung cấp một số chức năng bổ sung hơn so với NLTK.

TextBlob là một thư viện python và cung cấp một API đơn giản để truy cập các phương thức của nó và thực hiện các tác vụ NLP (Natural Language Processing) cơ bản. Một điều tốt về TextBlob là chúng giống như các chuỗi python. Vì vậy, có thể biến đổi và thao tác với nó giống như chúng ta đã làm trong python.

2.2.1. Một số thao tác phân tích văn bản bằng TextBlob:

2.2.1.1. Mã thông báo:

Mã thông báo đề cập đến việc phân chia văn bản hoặc một câu thành một chuỗi các mã thông báo, tương ứng với các từ ngữ. Đây là một trong những nhiệm vụ cơ bản của NLP. Để thực hiện việc này bằng TextBlob, có hai bước:

- ✓ Tạo một đối tượng textblob và truyền một chuỗi với nó.
- ✓ Gọi các chức năng của textblob để thực hiện một nhiệm vụ cụ thể.

```
from textblob import TextBlob


blob = TextBlob("Analytics Vidhya is a great platform to learn data science. \n It helps community th
rough blogs, hackathons, discussions,etc.")

blob.sentences
>> [Sentence("Analytics Vidhya is a great platform to learn data science."),
Sentence("It helps community through blogs, hackathons, discussions,etc.")]


blob.sentences[0]  ## extracting only first sentence
>> Sentence("Analytics Vidhya is a great platform to learn data science.")
```

Hình 2.1. Mô tả việc phân chia văn bản thành từng câu.

2.2.1.2. Gắn thẻ một phần của văn bản:

Gắn thẻ một phần của văn bản hoặc gắn thẻ ngữ pháp là một phương pháp để đánh dấu các từ có trong văn bản trên cơ sở định nghĩa và ngữ cảnh của nó. Nói một cách đơn giản, nó cho biết một từ là danh từ, tính từ hay động từ, v.v ...

```
for words, tag in blob.tags:
 print (words, tag)
>> Analytics NNS
Vidhya NNP
is VBZ
a DT
great JJ
platform NN
to TO
learn VB
data NNS
science NN
```

Hình 2.2. Ví dụ về phân loại từ; là danh từ, động từ hay tính từ.

Trong ví dụ trên, NN là danh từ, VB là động từ, JJ là tính từ,... và một số loại từ khác nữa.

2.2.1.3. N-grams:

Một sự kết hợp của nhiều từ với nhau được gọi là N-Grams. N gram (N>1) thường có nhiều thông tin hơn so với các từ và có thể được sử dụng làm các tính năng cho mô hình hóa ngôn ngữ. N-grams có thể dễ dàng truy cập trong TextBlob bằng cách sử dụng hàm ngrams, trả về một *tuple* của n từ liên tiếp.

```
for ngram in blob.ngrams(2):
print (ngram)
>> ['Analytics', 'Vidhya']
['Vidhya', 'is']
['is', 'a']
['a', 'great']
['great', 'platform']
['platform', 'to']
['to', 'learn']
['learn', 'data']
['data', 'science']
```

Hình 2.3. Ví dụ về N-grams.

2.2.1.4. Phân tích tình cảm:

Phân tích tình cảm về cơ bản là quá trình xác định thái độ hoặc cảm xúc của một người, tức là xem xem đó là tích cực, tiêu cực hay trung tính.

Hàm tình cảm của textblob trả về hai thuộc tính, *polarity* (phân cực) và *subjectivity* (tính chủ quan hay trung tính).

Phân cực là giá trị *float* nằm trong phạm vi [-1,1], trong đó 1 có nghĩa là tích cực và -1 có nghĩa là tiêu cực. Các câu chủ quan thường đề cập đến ý kiến cá nhân, cảm xúc hoặc phán đoán trong khi khách quan đề cập đến thông tin thực tế. Tính chủ quan cũng là một giá trị *float* nằm trong phạm vi [0,1].

```
print (blob)

blob.sentiment

>> Analytics Vidhya is a great platform to learn data science.

Sentiment(polarity=0.8, subjectivity=0.75)
```

Hình 2.4. Ví dụ về phân tích tính tiêu cực, tích cực bằng textblob.

Qua ví dụ trên, chúng ta có thể thấy rằng *polarity* là 0,8 có nghĩa là tích cực và 0,75 *subjectivity* cho thấy chủ yếu đó là một ý kiến cá nhân chứ không phải là một thông tin thực tế.

2.2.2. Những đặc tính thú vị khác:

2.2.2.1. Sửa lỗi chính tả:

Sửa lỗi chính tả là một tính năng thú vị mà TextBlob cung cấp, chúng ta có thể được truy cập bằng cách sử dụng chức năng *correct* như ví dụ sau.

```
blob = TextBlob('Analytics Vidhya is a gret platfrm to learn data scence')

blob.correct()

>> TextBlob("Analytics Vidhya is a great platform to learn data science")
```

Hình 2.5. Ví dụ về tính năng sửa lỗi chính tả của textblob.

2.2.2.2. Phát hiện ngôn ngữ và dịch thuật:

Cho một văn bản sử dụng loại ngôn ngữ sau:

هذا بارد

Với việc sử dụng textblob, ta hoàn toàn có thể biết được văn bản trên sử dụng ngôn ngữ nào. Chỉ với một câu lệnh *blob.detect_language()* như sau:

```
blob.detect_language()

>> 'ar'
```

Hình 2.6. Phát hiện ngôn ngữ với textblob.

Vậy, văn bản đó sử dụng ngôn ngữ Ả Rập để viết. Bây giờ, dịch nó sang tiếng Anh để chúng ta có thể hiểu những gì được viết bằng TextBlob.

```
blob.translate(from_lang='ar', to ='en')

>> TextBlob("that's cool")
```

Hình 2.7. Dịch văn bản sang tiếng Anh sử dụng textblob.

Ngay cả khi không xác định rõ ràng ngôn ngữ nguồn, TextBlob sẽ tự động phát hiện ngôn ngữ và dịch sang ngôn ngữ mong muốn.

Chương 3: PHÂN TÍCH VÀ THIẾT KẾ

3.1. Phân tích:

3.1.1. Tại sao lại là mạng xã hội?

Mạng xã hội (Social Networking Service – SNS) là nơi nối kết các thành viên cùng sở thích trên Internet lại với nhau với nhiều mục đích khác nhau không phân biệt không gian và thời gian. Những người tham gia vào dịch vụ mạng xã hội còn được gọi là cư dân mạng.

Mạng đổi mới hoàn toàn cách cư dân mạng liên kết với nhau và trở thành một phần tất yếu của mỗi ngày cho hàng trăm triệu thành viên khắp thế giới. Nó đã trở nên phổ biến hơn bao giờ hết, và chắc chắn trong tương lai mạng xã hội vẫn sẽ còn là nơi rất thu hút mọi người.

Sở dĩ chọn khai phá và phân tích trên mạng xã hội, là bởi vì:

Mạng xã hội là rất phổ biến: Ngày nay, nó đã xuất hiện trong rất nhiều lĩnh vực: xã hội, công nghệ thông tin, khoa học hành vi, toán học, thống kê và nhiều lĩnh vực khác nữa.

Mạng xã hội là kho tài nguyên tiềm năng khổng lồ: Nếu biết cách khai phá dữ liệu trên mạng xã hội, thì đây chính là một kho dữ liệu vô cùng lớn, là Big Data. Lấy ví dụ đối với mạng xã hội Facebook, theo thống kê vào cuối tháng 7/2018 thì trên thế giới có khoảng 2.196 triệu người sử dụng. Hãy thử tưởng tượng xem, nếu như với số lượng người như thế thì ta có thể khai thác được bao nhiêu thông tin?

3.1.2. Sự đối lập trong suy nghĩ của mỗi người:

Cuộc sống luôn tồn tại hai mặt đối lập nhau, như cái ác và cái thiện, cái xấu và cái tốt, người tham vọng và người luôn bằng lòng với cuộc sống mình đang có,... Rất chính xác, cuộc sống tồn tại hai mặt đối lập và suy nghĩ của con người cũng vậy, ngay thái độ của con người cũng có hai mặt đối lập. Giả sử khi sử dụng mạng wifi chẳng hạn, với tốc độ 30Mbps, có người sử dụng sẽ thấy nhanh nhưng

lại có người cảm thấy nó đang chạy với tốc độ "rùa bò". Như thế là đã hình thành hai luồng suy nghĩ trái ngược rồi.

Trên thực tế, đối với bất cứ một sự việc hay thậm chí là một con người nào đó, bản thân mỗi người đều có suy nghĩ riêng của mình, và thông thường nó chia làm hai chiều hướng: tích cực và tiêu cực.

Mạng xã hội cũng không thể tránh khỏi có những suy nghĩ trái chiều như vậy. Điển hình là phần bình luận của mỗi cá nhân đối với một bài viết nào đó. Những phản hồi tích cực thì không cần phải bàn tới, nhưng bên cạnh đó những phản hồi tiêu cực còn ẩn chứa rất nhiều mặt trái như sự ngộ độc thông tin, thông tin xấu, thông tin sai tràn lan, không thể kiểm soát,...

Internet đã tạo ra một không gian nơi thông tin sẽ được lan truyền nhanh hơn nhiều khi so với WOM (Word Of Mouth) truyền thống. Thông tin một cửa hàng bán đồ ăn nhanh không đảm bảo vệ sinh, có thể chỉ cần 15 phút để hàng triệu người biết đến, chúng sẽ nhanh chóng được chia sẻ trên các phương tiện truyền thông, đủ để mọi người tẩy chay ngay lập tức cửa hàng đó. Trong khi eWOM (Electronic Word Of Mouth) tích cực có thể tăng cường tiếp xúc, xây dựng lòng trung thành và tạo ra doanh thu, thì eWOM tiêu cực có thể là một thảm họa. Quản lý kịp thời các thông tin phản hồi hoặc bình luận tiêu cực sẽ giúp ta ngăn chặn nó trở thành thảm họa.

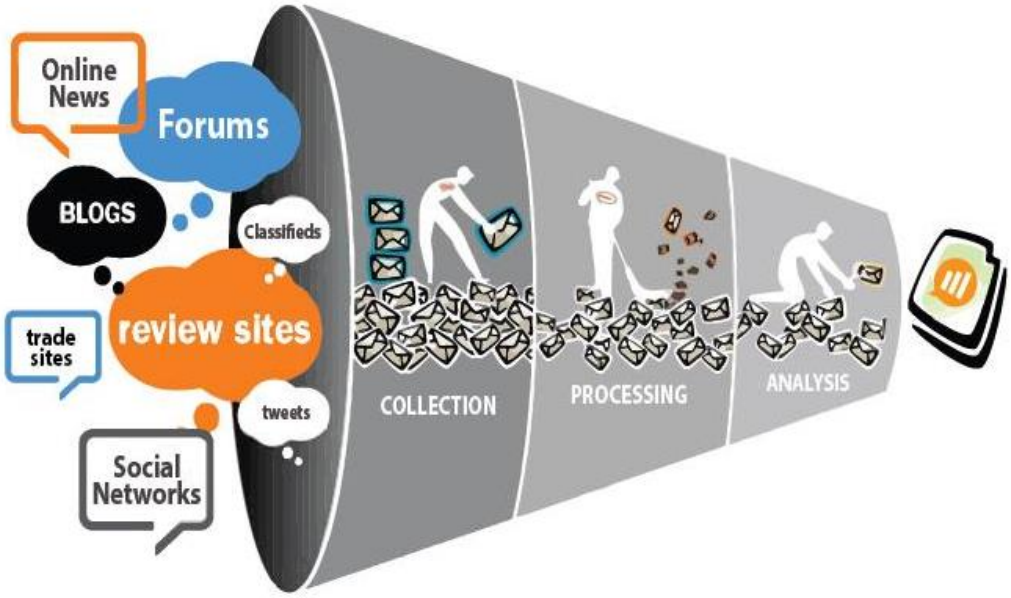
Nhưng trước khi có thể ngăn chặn được điều đó, bước đầu tiên ta cần phải lấy được những phản hồi của mọi người và tiến hành phân tích nói, thì mới có thể biết được đâu là phản hồi tích cực và đâu là phản hồi tiêu cực để mà đề phòng, ngăn chặn.

3.1.3. Hướng thực hiện:

Cũng giống như các quy trình nghiên cứu thị trường truyền thống, về cơ bản phải trải qua 5 giai đoạn:

- ✓ Thu thập dữ liệu.
- ✓ Xuất dữ liệu.

- ✓ Phân loại dữ liệu.
- ✓ Phân tích dữ liệu.
- ✓ Trình bày báo cáo nghiên cứu.



Hình 3.1. Mô hình quá trình nghiên cứu dữ liệu.

Trong đó, "Thu thập dữ liệu" là bước đầu tiên trong quá trình thực hiện đề tài. Hiện nay có hai phương pháp chính dùng để thu thập dữ liệu: API và Trang (Sites). Trong báo cáo này sẽ đề cập đến cách thu thập dữ liệu bằng cổng giao thức lập trình – API.

Phương pháp này được áp dụng đối với các *global social networks* như Facebook, Google Plus, Youtube, Twitter, Instagram,... trong đó các công cụ *social listening* sẽ kết nối với các API (Application Programming Interface – Giao diện lập trình ứng dụng) của các *social networks* và yêu cầu hệ thống trả về những bài viết có chứa *keywords*. Phương pháp này theo nguyên tắc cho phép lấy dữ liệu của toàn bộ *social network,* bao gồm các trang cá nhân, nhưng trên thực tế phụ thuộc vào sự hạn chế của các *social networks* này. Với việc Facebook hạn chế *organic reach* cho các chủ fanpage và các nhà quảng cáo, Facebook cũng không trả lại đầy đủ và nhất quán các bài viết cá nhân qua API. Hiện tại không có một

thống kê rõ ràng việc lấy dữ liệu bằng API có thể lấy được bao nhiêu % các thảo luận, phản hồi.

3.2. Thiết kế:

3.2.1. Các thư viện cần dùng:

Một chương trình để chạy được nhất định cần phải có các thư viện hỗ trợ. Chương trình thiết kế cho đề tài "Gom nhóm các nội dung tích cực, tiêu cực theo chủ đề trên mạng xã hội" này cũng cần có một số thư viện hỗ trợ. Vì chương trình viết bằng ngôn ngữ Python và làm việc với mạng xã hội Twitter nên có một số thư viện đặc thù.

Cụ thể:

```
import re, math, openpyxl, os
import tweepy,platform,time
from tweepy import OAuthHandler
from textblob import TextBlob
from openpyxl import Workbook
```

Hình 3.2. Các thư viện cần dùng đến trong chương trình.

Có một số thư viện đã quá quen thuộc với người lập trình như *math* và *time*, *math* hỗ trợ cho các phép tính toán số học và *time* có liên quan đến thời gian. Bên cạnh đó, để chương trình hoạt động còn cần cài đặt thêm các thư viện, module khác là: *platform*, *os*, *tweepy*, *textblob* và *openpyxl*. Tác dụng cụ thể của các thư viện, module này như sau:

- ✓ Platform: module *platform* trong Python được sử dụng để truy cập dữ liệu nền tảng cơ bản, chẳng hạn như: phần cứng và hệ điều hành.
- ✓ Os: module *os* trong Python cung cấp cách sử dụng chức năng phụ thuộc hệ điều hành.
- ✓ Tweepy: là thư viện Python giúp dễ dàng sử dụng trong việc truy cập Twitter API.
- ✓ Textblob: là một thư viện Python (2 và 3) để xử lý dữ liệu dạng văn bản.

- ✓ Openpyxl: là một thư viện Python hỗ trợ việc đọc/ghi các loại tệp xlsx/xlsm/xltx/xltm của Excel 2010.

3.2.2. Code thực hiện:

Phần này sẽ giải thích chi tiết về ý nghĩa của các đoạn code có trong chương trình.

```
class TwitterClient(object):
    '''
    Generic Twitter Class for sentiment analysis.
    '''
    def Create_Dir(self, dir_name):
            if not os.path.exists("data"):
                try:
                    os.mkdir("data")
                    print("Created directory 'data'")
                except:
                    print("Unable to create directory 'data': Directory already exists")
            else:
                print("Unable to create directory 'data': Directory already exists")

            if not os.path.exists("data/data_" + dir_name):
                try:
                    os.mkdir("data/data_" + dir_name)
                    print("Created directory 'data/data_" + dir_name + "'")
                except:
                    print("Unable to create directory 'data/data_" + dir_name + "': Directory already exists")
            else:
                print("Unable to create directory 'data/data_" + dir_name + "': Directory already exists")
             # Adding path.
            if not os.getcwd() in os.get_exec_path():
            # print('adding path')
                if platform.system() == "Windows":
                    os.environ["PATH"] = os.environ["PATH"] + ";" + os.getcwd()
                else:
                    os.environ["PATH"] = os.environ["PATH"] + ":" + os.getcwd()
```

Hình 3.3. Code thực hiện chức năng tạo thư mục lưu trữ dữ liệu.

Hàm *Create_Dir(self, dir_name)* dùng để tạo thư mục lưu trữ dữ liệu sau khi được trích xuất. Trong hàm này sẽ thực hiện tạo ra hai thư mục: thư mục cha có tên là *data* và thư mục con có dạng *data_dir_name*. Đối với thư mục *data*, đầu tiên sẽ tiến hành kiểm tra xem thư mục này đã tồn tại hay chưa: nếu chưa thì tạo một thư mục mới, ngược lại sẽ in ra câu thông báo “Unable to create directory ‘data’: Directory already exists”. Tương tự khi tạo thư mục con trong thư mục

data cũng vậy, điểm khác biệt ở đây là tên của thư mục con. Tên thư mục khi tạo ra sẽ có dạng *data_dir_name* (ví dụ: data_love, data_dog,...), với *dir_name* là *keyword* (như trong ví dụ trước đó, *keyword* là love và dog) mà người dùng muốn tìm kiếm.

Cuối cùng là thêm đường dẫn *Path* vào cho các thư mục vừa tạo.

```
def __init__(self):
    '''
    Class constructor or initialization method.
    '''
    # keys and tokens from the Twitter Dev Console
    consumer_key = 'Y2x9rnSVirom9p4aP4j4GmWGy'
    consumer_secret = 'peCyxeqB68YvMrcdjXilHbzbWe3KXZKD4cE6NzHnjxTd2GCm9u'
    access_token = '2186053585-zX6VlzWtTr9nNg72SXk9q0TWe6yV6VDyI0TCaxF'
    access_token_secret =  'T3NAV6vXeOzHXLwBRulUXyBxRQUP8cjdbepFkeFzQyMgh'

    # attempt authentication
    try:
        # create OAuthHandler object
        self.auth = OAuthHandler(consumer_key, consumer_secret)
        # set access token and secret
        self.auth.set_access_token(access_token, access_token_secret)
        # create tweepy API object to fetch tweets
        self.api = tweepy.API(self.auth)
    except:
        print("Error: Authentication Failed")

def clean_tweet(self, tweet):
    '''
    Utility function to clean tweet text by removing links, special characters
    using simple regex statements.
    '''
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())
```

Hình 3.4. Code thực hiện khởi tạo thông tin truy cập và xóa những thứ không cần thiết.

Hàm *_init_(self)* dùng để khởi tạo tweepy API để có thể truy cập vào một trang Twitter và lấy dữ liệu từ trang đó. Các thông tin này bao gồm: *consumer_key*, *consumer_secret*, *access_token* và *access_token_secret*. Sau khi đã

có các thông tin trên, ta sẽ tiến hành xác thực. Trước hết cần phải tạo một đối tượng *OauthHandler*, tiếp đó là đặt mã thông báo và mã thông báo bí mật (*acces_token* và *access_token_secret*) và cuối cùng là tạo đối tượng tweepy API để lấy các tweet trên Twitter. Nếu các thao tác trên không thực hiện được sẽ thông báo lỗi "Error: Authentication Failed".

Đối với hàm *clean_tweet(self, tweet)*, chức năng tiện ích này để làm sạch văn bản tweet bằng cách xóa các liên kết, ký tự đặc biệt bằng cách sử dụng các câu lệnh *regex* đơn giản.

```
def get_tweet_sentiment(self, tweet):
    '''
    Utility function to classify sentiment of passed tweet
    using textblob's sentiment method
    '''
    # create TextBlob object of passed tweet text
    analysis = TextBlob(self.clean_tweet(tweet))
    # set sentiment
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'
```

Hình 3.5. Code thực hiện công việc phân tích văn bản.

Hàm *get_tweet_sentiment(self, tweet)* chủ yếu phân tích các dữ liệu văn bản lấy được xem xem đó là tích cực, tiêu cực hay trung tính. Để làm được điều này, cần có sự hỗ trợ của textblob. Đầu tiên tạo một đối tượng textblob của văn bản tweet đã lấy được. Sau đó tiến hành phân tích, nếu phân cực (*polarity*) mà lớn hơn giá trị 0 thì văn bản đó là tích cực (*positive*), nếu như có giá trị bằng 0 đó sẽ là trung tính (*neutral*) và ngược lại của hai trường hợp trên là tiêu cực (*negative*).

```
def get_tweets(self, query, count = 1000,lang='en'):
    '''
    Main function to fetch tweets and parse them.
    '''
    # empty list to store parsed tweets

    tweets = []
    try:
        # call twitter api to fetch tweets
        fetched_tweets = self.api.search(q = query, count = count)
        # parsing tweets one by one
        for tweet in fetched_tweets:
            # empty dictionary to store required params of a tweet
            parsed_tweet = {}

            # saving text of tweet
            parsed_tweet['text'] = tweet.text
            # saving sentiment of tweet
            parsed_tweet['sentiment'] = self.get_tweet_sentiment(tweet.text)

            # appending parsed tweet to tweets list
            if tweet.retweet_count > 0:
                # if tweet has retweets, ensure that it is appended only once
                if parsed_tweet not in tweets:
                    tweets.append(parsed_tweet)
            else:
                tweets.append(parsed_tweet)

        # return parsed tweets
        return tweets
    except tweepy.TweepError as e:
        # print error (if any)
        print("Error : " + str(e))
```

Hình 3.6. Code thực hiện chức năng lấy ra các tweets.

Hàm *get_tweets(self, query, count = 1000, lang = 'en')* thực hiện chức năng trích xuất các tweet, cho phép lấy ra 1000 dòng dữ liệu và ngôn ngữ sử dụng ở đây là Tiếng Anh.

Trước hết cần tạo một danh sách trống để lưu trữ các tweet sau khi được phân tích. Tiếp đó tiến hành các thao tác xử lý chính trong khối *try – except*. Đầu tiên là gọi twitter API để lấy các tweets. Trong vòng lặp *for* sẽ xử lý các tweets đã lấy được đó theo từng dòng một. Lưu lại văn bản và tình cảm (*sentiment*) của các tweet, cuối cùng là đẩy các tweet đã được phân tích cú pháp vào trong danh sách tweet (*parsed_tweet)*.

Nếu có bất kỳ lỗi nào thì sẽ in ra câu thông báo bị lỗi.

```
def main():
    # creating object of TwitterClient Class
            api = TwitterClient()
    # calling function to get tweets
            query = str(input("Enter keyword to search for: "))
            tweets = api.get_tweets(query, count = 1000, lang = 'en')

            api.Create_Dir(query)
            time.sleep(5)
            print("\n\nStarting Scrapping Twitter")

            file_path = "data/data_" + query
           # Create a workbook for excel
            tag_File = file_path + "/" + query + "_Twitter.xlsx"

            wb = openpyxl.Workbook()
            ws_Pos = wb.create_sheet(title="Positive")
            col = 'A'
            row = 1

    # picking positive tweets from tweets
            ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
            print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
            time.sleep(5)
    # picking negative tweets from tweets
            ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
            print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
            time.sleep(5)
    # percentage of neutral tweets
            nuatral = (len(tweets) - len(ntweets))- len(ptweets)
            print("Neutral tweets percentage: {} % ".format(100*(len(tweets) - len(ntweets)- len(ptweets))/len(tweets)))
            time.sleep(5)
    # printing  positive tweets
            print("\n\nPositive tweets:")
            for tweet in ptweets:
                print(tweet['text'])
                ws_Pos['A' + str(row)] = tweet['text']
                row += 1
            time.sleep(5)
    # printing  negative tweets
            ws_Neg = wb.create_sheet(title="Negative")
            col = 'A'
            row = 1
            print("\n\nNegative tweets:")
            for tweet in ntweets:
                print(tweet['text'])
                ws_Neg['A' + str(row)] = tweet['text']
                row += 1
            time.sleep(5)

            wb.save(tag_File)
```

Hình 3.7. Code hàm *main()*.

Trong hàm *main()*, trước hết tạo một đối tượng TwitterClient Class (các hàm trong các hình *3.3*, *3.4*, *3.5*, *3.6* đều nằm trong class TwitterClient) có tên là *api*. Sau đó, gọi hàm chức năng để lấy các tweet ra; trước đó tạo một query in ra câu “Enter keyword to search for: ” và yêu cầu người dùng nhập vào từ khóa.

Lại gọi tiếp hàm *Create_Dir(query)* để tạo thư mục *data* chứa các dữ liệu trích xuất được với thời gian nghỉ là 5 giây. Trong thời gian chờ đó in ra câu "Starting Scrapping Twitter" hiện trên màn hình. Tiếp đó tạo một file Excel có dạng *query_Twitter* nằm trong thư mục *data_query* (*query* là từ khóa nhập vào lúc đầu).

Nếu phân tích tình cảm đoạn văn bản của một tweet là *positive* thì in ra câu "Positive tweets percentage: {} %", trong đó tỉ lệ phần trăm này được tính bằng số tweet *positive* trên tổng số tweet. Tương tự đối với tweet là *negative*, còn nếu tweet là *neutral* thì tỉ lệ phần trăm sẽ lấy 100 % trừ đi tỉ lệ của *positive* và *negative*.

Cuối cùng là gom nhóm các tweet cùng loại lại với nhau. Nếu tweet là *positive* thì in ra tweet đó trên màn hình, đồng thời đẩy vào trong file Excel ở Sheet *Positive* và tăng số dòng (*row*) lên 1 đơn vị. Thực hiện thao tác tương tự với các tweet là *negative*.

Sau cùng sẽ lưu lại file và kết thúc chương trình.

Chương 4: KẾT QUẢ THỰC NGHIỆM

4.1. Cài đặt:

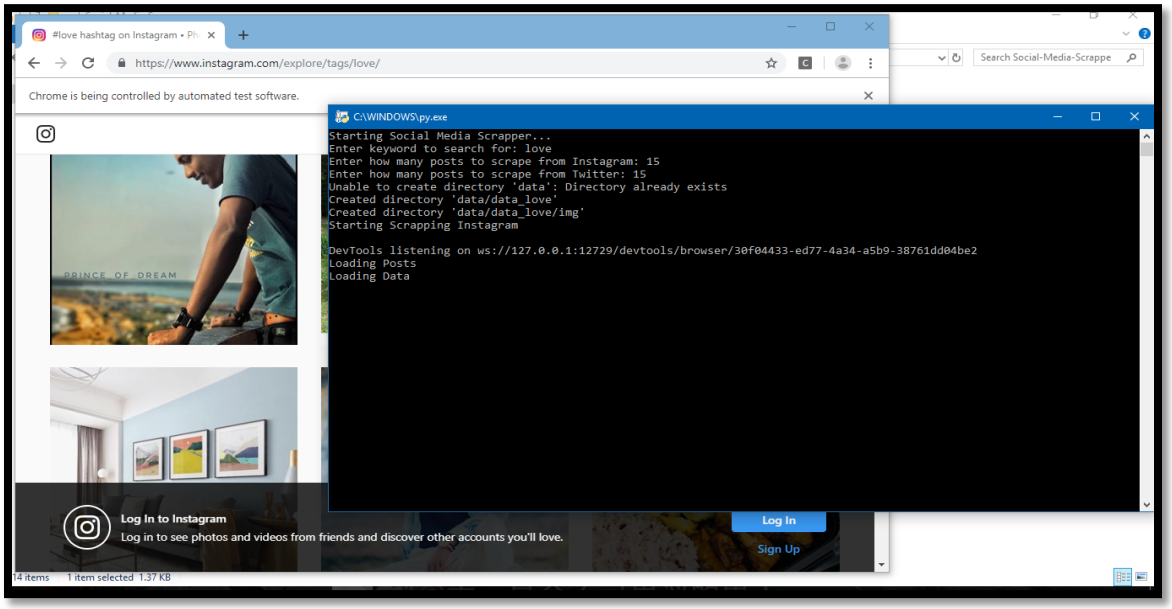
Yêu cầu phần mềm: Phải trích xuất được tất cả các phản hồi, bình luận có liên quan; đồng thời phân tích, đánh giá độ tích cực, tiêu cực của các phản hồi trên mạng xã hội.

Yêu cầu phần dữ liệu: Trích xuất dữ liệu và xuất thành một file Excel. Trên đó có thể thấy được các phản hồi, bình luận đã lấy được trên mạng xã hội.

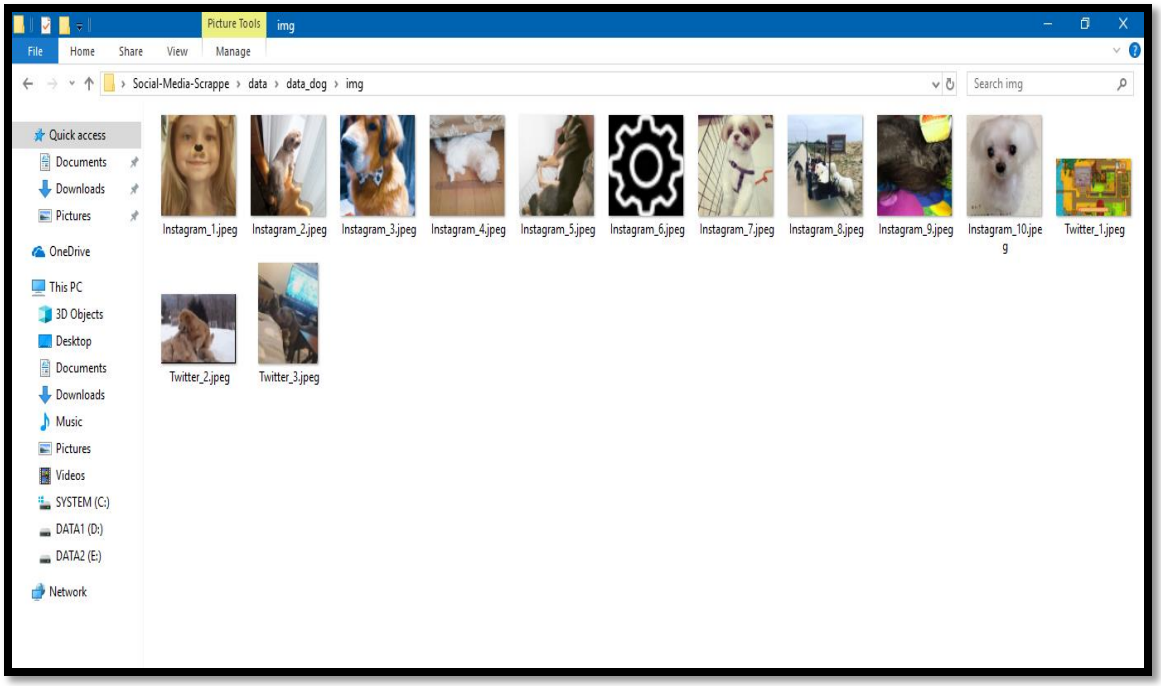
4.2. Thử nghiệm:

Trong quá trình thực hiện, việc mắc sai lầm là không thể tránh khỏi. Bước đầu tiên khi mới bắt tay vào lập trình, còn chưa biết nên bắt đầu từ đâu vì vậy mà cài đặt lỗi vô số lần. Khi đã có thể chạy được lại trích xuất ra dữ liệu không như mong muốn hoặc phân tích phản hồi tích cực, tiêu cực không được chính xác,...

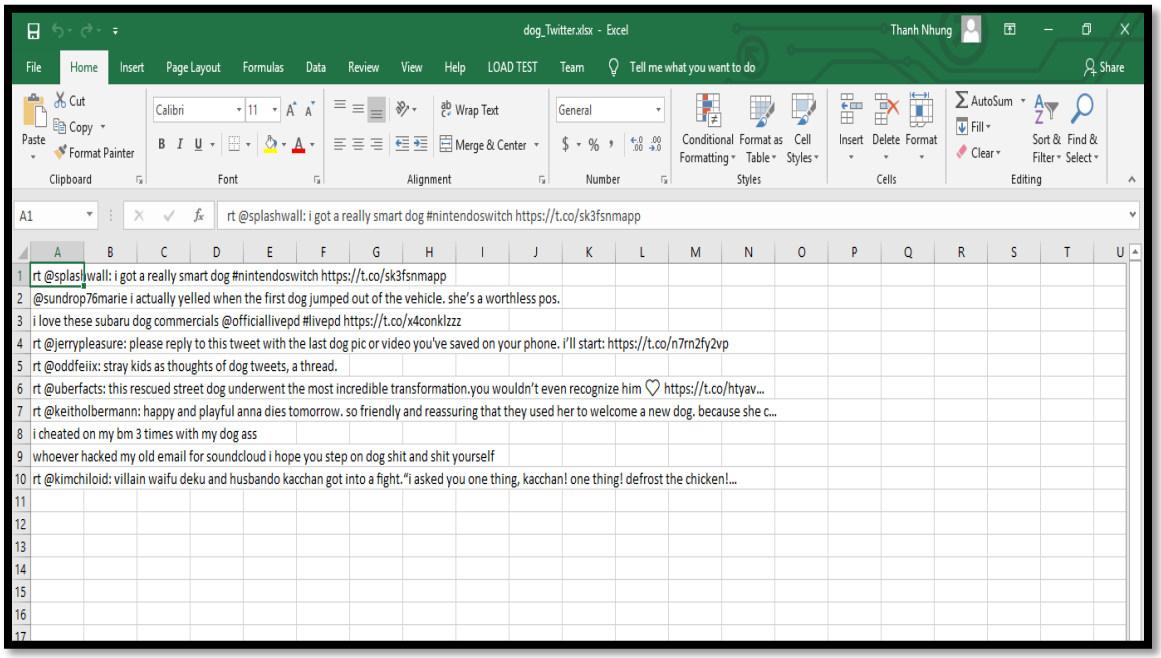
Sau đây là một số bản chạy thử đối với hai mạng xã hội là Instagram và Twitter:



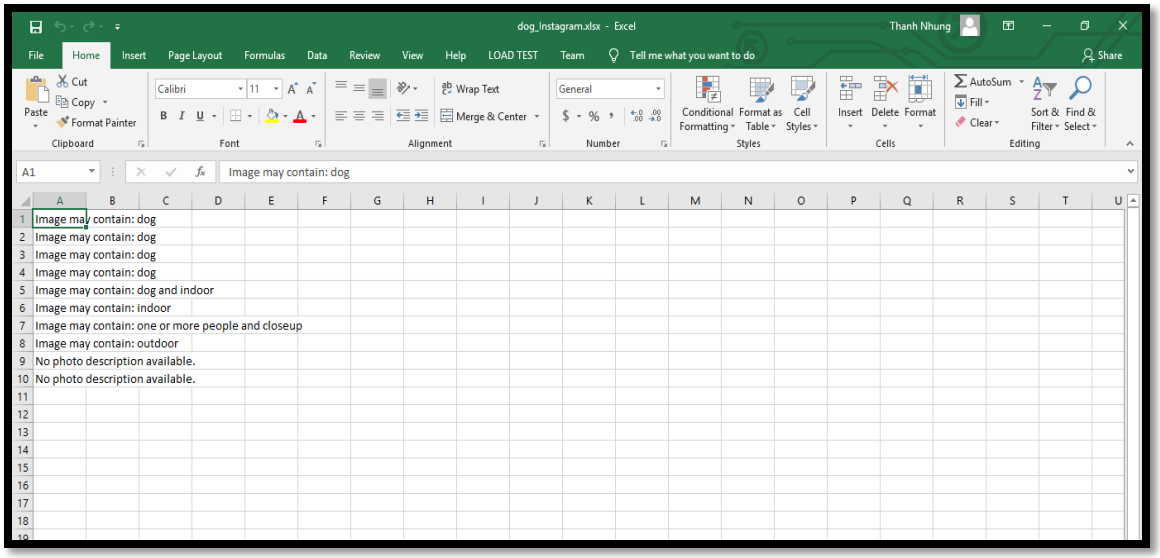
Hình 4.1. Chạy lệnh trên py.exe: nhập vào từ khóa tìm kiếm và số lượng bài đăng muốn lấy lần lượt trên Instagram và Twitter.



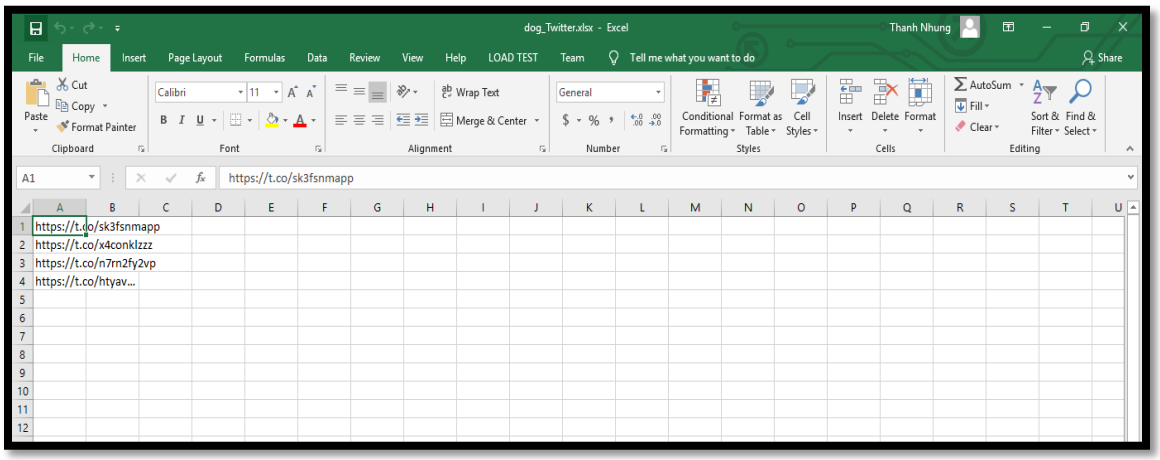
Hình 4.2. Các hình ảnh lấy về được lưu vào trong một thư mục có tên là data.



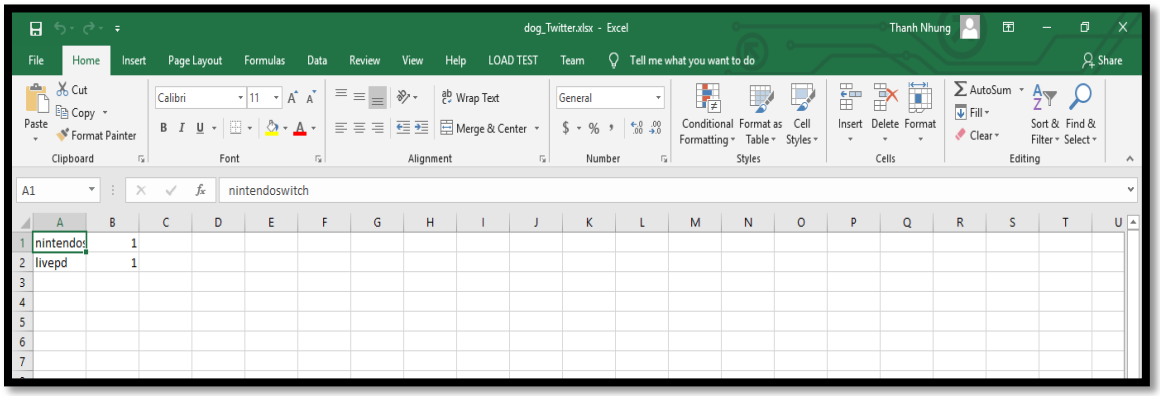
Hình 4.3. Nội dung của mỗi bài đăng được lấy về và xuất ra file Excel với tên Sheet là Caption đối với mạng xã hội Twitter.



Hình 4.4. Các bài đăng có hình ảnh có thể chứa từ khóa trên Instagram.



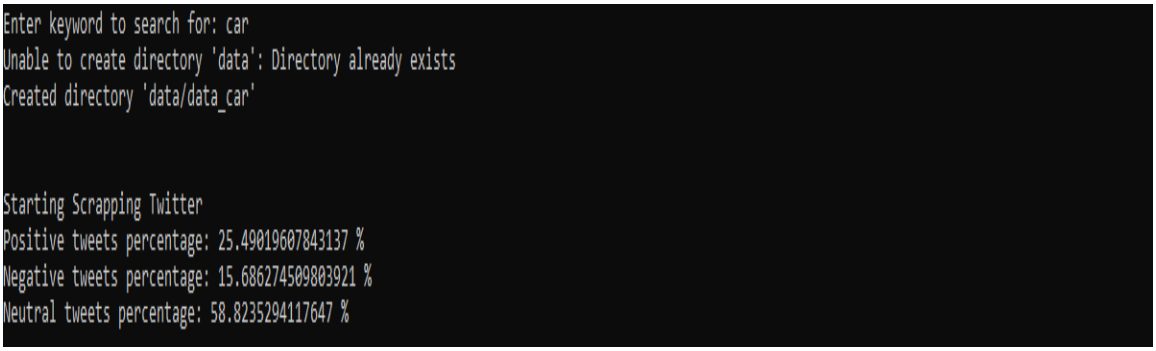
Hình 4.5. Các đường link lấy được tương tự cũng sẽ lưu vào Sheet Links.



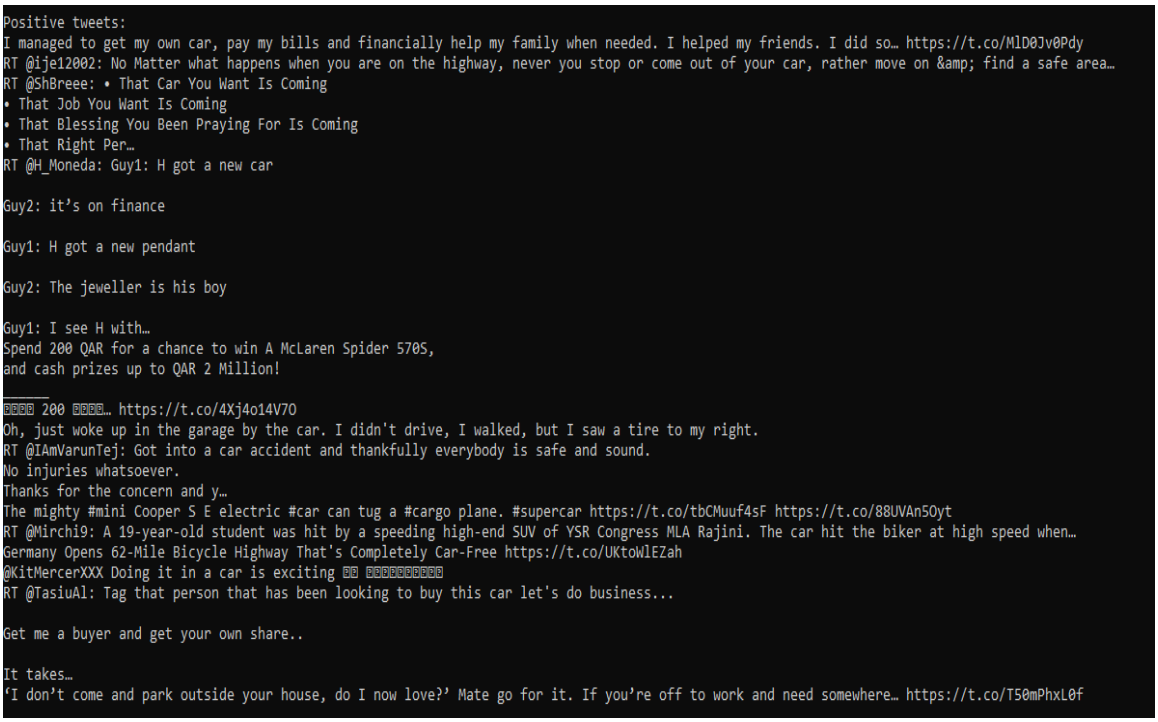
Hình 4.6. Đối với các tag thì lưu vào Sheet Tags.

Những hình ảnh trên được lấy từ một lần chạy thử chương trình. Tuy nhiên, sau này đã không được tiếp tục sử dụng. Sở dĩ không phát triển tiếp từ chương trình này bởi vì sau này hướng chủ đạo của đề tài là tập trung vào mạng xã hội Twitter nên Instagram không cần thiết. Ngoài ra, những ví dụ trên không cho phép trích xuất những phản hồi, bình luận của người dùng cho nên không phù hợp với yêu cầu của đề tài.

Sau đây là kết quả thực thi chương trình hoàn chỉnh:



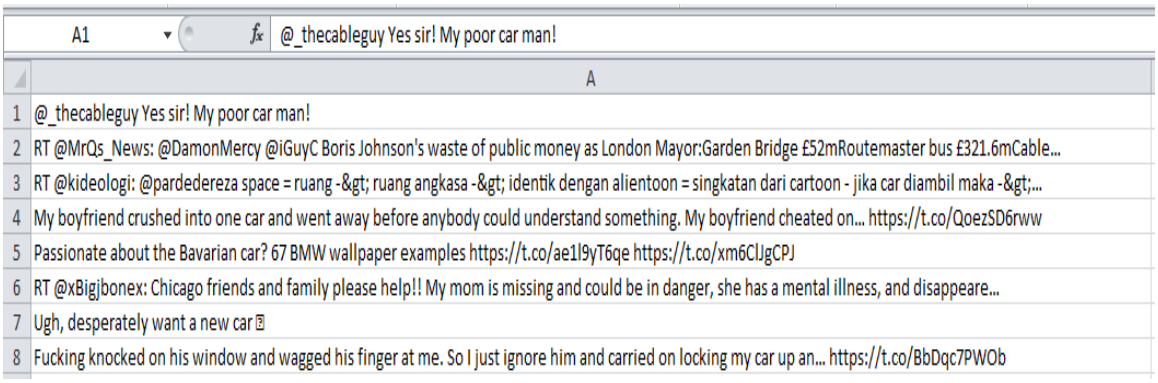
Hình 4.7. Nhập vào từ khóa tìm kiếm, chương trình sẽ tạo thư mục lưu trữ và tính toán tỉ lệ phân tích tình cảm của các tweet.



Hình 4.8. Gom nhóm các tweet *positive*.

Hình 4.9. Gom nhóm các tweet nega*tive*.

Hình 4.10. Sheet Positive lưu các tweet mang tính tích cực.



Hình 4.11. Sheet Negative lưu các tweet mang tính tiêu cực.

Chương 5: TỔNG KẾT

5.1. Các kết quả đã thực hiện:

5.1.1. Về yêu cầu của đề tài:

Kết quả cuối cùng thực hiện được đáp ứng được các yêu cầu cơ bản sau:

- ✓ Cho phép truy xuất dữ liệu lấy được trên trang mạng xã hội Twitter: nội dung bài đăng và bình luận.
- ✓ Các dữ liệu lấy được được lưu trữ trong một thư mục có tên dưới dạng data_keyword (trong đó keyword là từ khóa muốn tìm kiếm). Các dữ liệu lấy được lưu trữ vào các Sheet trong file Excel. Ví dụ: các phản hồi, bình luận mang tính tích cực thì được lưu vào Sheet Positive,...
- ✓ Phân tích các phản hồi, bình luận lấy được và xem xét xem chúng là tích cực hay tiêu cực, sau đó gom nhóm chúng lại với nhau.

5.1.2. Thu hoạch cá nhân:

Đối với cá nhân các thành viên trong nhóm lập trình, qua lần thực hiện đề tài này cũng có những thu hoạch nhất định:

- ✓ Hiểu biết thêm về ngôn ngữ lập trình Python và những gói, thư viện đi kèm như: TextBlob, BeautifulSoup, Tweepy,...
- ✓ Có cái nhìn khách quan hơn trong việc sử dụng mạng xã hội; biết cân nhắc mỗi khi đăng tải một phản hồi, một bình luận nào đó.
- ✓ Có cơ hội tiếp xúc thêm với nhiều mạng xã hội lớn khác ngoài Facebook, như: Twitter, Instagram, Wechat, Weibo,...
- ✓ Rèn luyện được kỹ năng viết báo cáo, kỹ năng lập trình và kỹ năng làm việc nhóm.

5.2. Đánh giá ưu, khuyết điểm:

5.2.1. Ưu điểm:

Kết quả thực hiện được có một số ưu điểm đáng xem xét sau:

- ✓ Thời gian truy xuất dữ liệu nhanh.

- ✓ Gom nhóm các phản hồi, bình luận một cách rõ ràng, không mang tính đại khái, chung chung.
- ✓ Có thể hỗ trợ trong việc phân tích, đánh giá của các doanh nghiệp kinh doanh, giúp cho họ có cái nhìn khách quan hơn về người tiêu dùng.
- ✓ Nguồn dữ liệu lấy được từ mạng xã hội là rất lớn, cho nên việc phân tích các phản hồi, bình luận được chính xác và có giá trị tham khảo cao.

5.2.2. Khuyết điểm:

Bên cạnh đó, cũng có những khuyết điểm cần được cân nhắc thêm:

- ✓ Việc truy xuất các dữ liệu thông qua API gặp một số trở ngại, do đó gặp một số hạn chế về vấn đề bảo mật và phân quyền.
- ✓ Hiện tại không có một thống kê rõ ràng việc lấy dữ liệu bằng API có thể lấy được bao nhiêu % các cuộc thảo luận trên mạng xã hội.

5.3. Hướng mở rộng trong tương lai:

Hiện tại thì chương trình chỉ dừng lại ở việc truy xuất dữ liệu ra và nó đều là bằng Tiếng Anh. Trong tương lai, chương trình sẽ phát triển thành truy xuất Tiếng Anh, sau đó chuyển sang Tiếng Việt rồi mới tiến hành việc phân tích gom nhóm các phản hồi đã lấy được.

Thêm vào đó sẽ nghiên cứu cách truy xuất dữ liệu từ mạng xã hội Facebook. Đồng thời tìm cách khắc phục nhược điểm về việc Facebook hạn chế *organic reach* cho các chủ fanpage và các nhà quảng cáo, Facebook cũng không trả lại đầy đủ và nhất quán các bài viết cá nhân qua API. Có thể cân nhắc đến tình huống sử dụng phương pháp thu thập dữ liệu theo Sites (Trang) để tránh tình huống này. Hiện nay ở nước ta, Facebook là một mạng xã hội lớn, vượt trội hơn so với các mạng xã hội. Nếu có thể lấy dữ liệu từ Facebook thì đây sẽ là nguồn dữ liệu vô cùng phong phú.

TÀI LIỆU THAM KHẢO

[1] https://congnghe.tuoitre.vn/nhip-song-so/hay-van-minh-khi-dung-mang-xa-hoi-1329711.htm

[2] https://vi.wikipedia.org/wiki/Python

[3] https://beau.vn/vi/goc-nhin/hai-mat-cua-social-media-marketing

[4] https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

[5] https://buzzmetrics.com/he%CC%A3-thong-thu-tha%CC%A3p-du%CC%83-lie%CC%A3u-cu%CC%89a-social-listening-tool-khu%CC%89ng-va-tinh-vi-den-muc-nao/?fbclid=IwAR06XQEYpLKZ1J6Z9RmuokpQEWTSc_V2xIwWuU8i7lThx2XC4K0oRs6FznE

[6] https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/

[7] https://vi.wikipedia.org/wiki/Twitter