ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN

THAM GIA XÉT GIẢI THƯỞNG "NHÀ KHOA HỌC TRẺ UEL"

NĂM 2020

Tên đề tài:

KHAI PHÁ Ý KIẾN NGƯỜI DÙNG THÔNG QUA TIN TỨC TRỰC TUYẾN TRONG LĨNH VỰC ĐIỆN THOẠI THÔNG MINH BẰNG MACHINE LEARNING

Lĩnh vực khoa học: Quản lý – Tin học

TP.HCM, Tháng 4 Năm 2020

ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN

THAM GIA XÉT GIẢI THƯỞNG "NHÀ KHOA HỌC TRẺ UEL"

NĂM 2020

Tên đề tài:

KHAI PHÁ Ý KIẾN NGƯỜI DÙNG THÔNG QUA TIN TỨC TRỰC TUYẾN TRONG LĨNH VỰC ĐIỆN THOẠI THÔNG MINH BẰNG MACHINE LEARNING

Nhóm sinh viên thực hiện

TT	Họ tên	MSSV	Đơn vị	Nhiệm vụ	Điện thoại	Email
1.	Trần Anh Thuận	K174111276	Khoa HTTT	Nhóm trưởng	0777021389	thuanta17411@st.uel.edu.vn
2.	Bùi Xuân Thành	K174060719	Khoa HTTT	Tham gia	0931528922	thanhbx17406c@st.uel.edu.vn
3.	Võ Nguyễn Tâm An	K174111293	Khoa HTTT	Tham gia	0819768308	anvnt17411c@st.uel.edu.vn
4.	Nguyễn Anh Nhật	K174111311	Khoa HTTT	Tham gia	0395963731	nhatna17411c@st.uel.edu.vn

Giảng viên hướng dẫn: **TS. Lê Hoành Sử**

TP.HCM, Tháng 4 Năm 2020

TÓM TẮT ĐỀ TÀI

- Khái quát về đề tài

Bài nghiên cứu khoa học này sẽ dựa vào các dữ liệu bài báo và bình luận thu thập được trang báo trực tuyến và sử dụng hai mô hình để phân loại tin tức trực tuyến, bao gồm dự đoán nội dung tin tức và dự đoán bình luận. Dựa trên đánh giá, nhận xét, phản hồi từ người dùng, kết quả nghiên cứu tạo ra những công cụ có thể nhận biết ý kiến và cảm xúc của người dùng thông qua những cuộc trao đổi trên các bản tin cũng như độ hiệu quả của nội dung các bản tin đó trong việc lôi kéo người dùng thảo luận.

- Mô tả một số phương pháp nghiên cứu chính của nghiên cứu.

Phương pháp thu thập thông tin.

Phương pháp phân tích và tổng hợp lý thuyết..

Phương pháp thống kê.

Phương pháp định lượng.

- Tóm lược các kết quả nghiên cứu đã đạt được và các nhận định chính.

Từ 2500 bài báo và 8700 bình luận, chúng tôi có kết quả phân loại tích cực, trung tính và tiêu cực. Trong đó, mô hình dự đoán tin tức hiệu quả nhất là BERT với F1 Score là 74% còn mô hình dự đoán nhận xét bình luận hiệu quả nhất là BiLSTM với 71%.

- Các kết luận và đề xuất chính:

Nghiên cứu đem lại kết quả khả quan và có tính ứng dụng tốt đi cùng với lĩnh vực tin tức trực tuyến ngày càng phát triển, tuy nhiên một số bước trong nghiên cứu còn thực hiện thủ công. Trong tương lai, các kết quả nghiên cứu có thể là công cụ tốt cho Marketing, R&D và phát triển sản phẩm trong công ty, hoặc ứng dụng cộng đồng như dự đoán fake news, một vấn đề quan trọng đối với thông tin mạng.

Mục lục

DANH SÁCH BẢNG

DANH SÁCH HÌNH

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu

Báo cáo của GFK trong ba quý đầu năm 2019 [24] cho thấy khối lượng thị trường của điện thoại thông minh tại Việt Nam là hơn 3 triệu đô la. Điều này chứng tỏ rằng người Việt Nam đang chi nhiều tiền hơn cho điện thoại thông minh và thay đổi mô hình thường xuyên hơn.

Ngoài ra, người dân Việt Nam cũng truy cập vào Internet thường xuyên. Theo báo cáo Digital Marketing Vietnam 2020 [9], người dùng điện thoại thông minh chiếm 93% tổng dân số tại Việt Nam. Họ dành 6 giờ 30 phút mỗi ngày trên Internet và 70% dân số (tương ứng 68.17 triệu người) sử dụng internet hàng ngày.

Song song với xu hướng này, tin tức trực tuyến cũng trở nên đa dạng và thú vị hơn. Hơn nữa, hiệu quả của những tin tức, đánh giá hoặc nhận xét này về suy nghĩ và hành vi của khách hàng rõ ràng đang tăng lên.

Trong nghiên cứu khoa học này, chúng tôi đã sử dụng hai mô hình để phân tích tổng thể một bản tin. Mô hình đầu tiên được sử dụng nhằm dự đoán nội dung bản tin đó và mô hình còn lại được sử dụng để dự đoán bình luận người dùng. Trong mỗi mô hình, chúng tôi chọn một phương pháp hiệu quả nhất trong số bốn phương pháp học máy (LSTM, BiLSTM, SVM và BERT). Cuối cùng, chúng tôi kết hợp đầu ra của hai mô hình.

1.2. Tính cấp thiết

Ngày nay, công nghệ ngày càng phát triển, đặc biệt với sự ra đời của Internet, lượng thông tin mà chúng ta tiếp cận ngày càng tăng lên theo cấp số nhân và thực sự

là một kho báu. Nhiệm vụ của chúng tôi là làm thế nào khai thác triệt để kho báu này cũng như biết cách lọc thông tin rác và giữ lại những dữ liệu cần thiết và trọng yếu.

Xu thế Internet đi cùng thị trường điện thoại thông minh này tạo cơ hội cho việc đánh giá sản phẩm giữa những người dùng với nhau trở nên dễ dàng và minh bạch hơn. Người dùng luôn quan tâm và bình luận, trao đổi về chất lượng sản phẩm và mong muốn có thêm nhiều ý kiến để tham khảo cho quyết định của họ, doanh nghiệp thì quan tâm nhiều hơn về các ý kiến đánh giá, ưu điểm cũng như khuyết điểm sản phẩm của mình từ chính khách hàng. Từ đó phát triển nhu cầu đòi hỏi phải có một công cụ tính toán sự ảnh hưởng của những nội dung tin tức truyền thông, quảng cáo, bài đánh giá, chính sách đối với sản phẩm điện thoại thông minh một cách hoàn toàn tự động. Vì thế, việc nghiên cứu và tìm hiểu về khai phá dữ liệu tin tức ứng dụng vào dự báo kinh tế nói chung và lĩnh vực điện thoại thông minh nói riêng là một định hướng nghiên cứu khoa học đầy hứa hẹn và thực tiễn, có tính ứng dụng cao và có thể dễ dàng mở rộng sang những lĩnh vực khác.

1.3. Nghiên cứu có liên quan

❖ Nghiên cứu trong nước:

(1) **PGS.TS Lê Anh Cường**, *Hệ thống thu thập và Phân tích quan điểm cộng đồng mạng đối với các sản phẩm thương mại*: xây dựng một hệ thống (đặt tại trang ooz.vn) thu thập tất cả các thông tin bình luận về sản phẩm điện thoại di động, sau đó phân tích để đưa cho người dùng thông tin tổng hợp về quan điểm khen chê đối với sản phẩm mà người dùng đang tìm kiếm. Từ đó người dùng sẽ có được quyết định phù hợp để mua sản phẩm.

(2) **Kiều Thanh Bình (2010),** *Tự động đánh giá quan điểm người dùng:* Nghiên cứu sử dụng công cụ để đánh giá một cách chính xác nhất về sản phẩm, đánh giá xem tính năng của sản phẩm này được mọi người tiếp đón thế nào.

(3) **Võ Tuyết Ngân, Đỗ Thanh Nghị (2015),** *Phân loại ý kiến trên TWITTER:* nhóm tác giả đề xuất sử dụng mô hình túi từ và giải thuật máy học Multinomial Naïve Bayes để phân loại ý kiến người dùng trên mạng xã hội Twitter.

(4) Cùng với một số nghiên cứu khác.

❖ Nghiên cứu có liên quan nước ngoài:

(1) **Ritu Mewari, Ajit Singh, Akash Srivastava (2015), Opinion** *Mining Techniques on Social Media Data:* Bài nghiên cứu của nhóm tác giả đưa ra và khái quát hóa các kỹ thuật cũng như hướng tiếp cận của quy trình khai phá quan điểm người dùng trên Mạng xã hội và so sánh những kỹ thuật khai phá ở thời điểm nghiên cứu.

(2) **Vignesh Rao, Jayant Sachdev (2017),** *A machine learning approach to classify news articles based on location:* Trong bài nghiên cứu này, nhóm tác giả đã sử dụng các mô hình Random Forest, Naive Bayes và SVM để phân loại tin tức Tiếng Anh cho từng cá nhân dựa trên thành phố mà họ đang sống, từ đó cung cấp cho họ một danh sách tin tức phù hợp nhất với từng cá nhân.

(3) **Yiou Lin, Hang Lei, Jia Wu, Xiaoyu Li,** *An Empirical Study on Sentiment Classification of Chinese Review using Word Embedding:* Nghiên cứu này các tác giả đã sử dụng kỹ thuật Word Embedding để phân loại cảm xúc của người viết thông qua các đánh giá bằng tiếng Trung Quốc trên Mạng xã hội, bài nghiên cứu tập trung sử dụng các mô hình như SVM, Logistic Regression, Convolutional Neural Network (CNN) và mô hình kết hợp.

(4) Cùng với một số nghiên cứu khác.

1.4. Các vấn đề

Với sự tăng trưởng về khối lượng dữ liệu Internet, đặc biệt là các review (đánh giá) trực tuyến (như đánh giá sản phẩm, đánh giá phim, v.v.), nhiều nghiên cứu hiện nay đang tập trung vào phân tích ý kiến, cũng được biết đến như khai phá ý kiến

người dùng. Đây là một lĩnh vực nghiên cứu mới bao gồm khai thác thông tin (IR), xử lý ngôn ngữ tự nhiên (NLP) và ngôn ngữ học máy tính. Hệ thống thường tìm các từ và cụm từ chính, phân tích ngữ cảnh, sau đó phân loại câu, đoạn hoặc toàn bộ tài liệu dựa trên những thông tin này. Khác với việc phân loại thể loại hoặc chủ đề, phân loại quan điểm đòi hỏi phải phân tích được cảm xúc của người dùng trong văn bản đó. Bên cạnh đó, khai phá ý kiến cũng tồn tại các thách thức liên quan đến xây dựng từ điển thuật ngữ, độ phức tạp của từ, từ trong ngữ cảnh khác nhau, từ đồng nghĩa... Cụ thể hơn, các câu hỏi nghiên cứu được đặt ra là:

- Làm sao mà doanh nghiệp có thể biết được người dùng có đang ưa chuộng sản phẩm của mình hay không? Đâu là thời điểm ra mắt sản phẩm mới?
- Doanh nghiệp làm sao có thể lôi kéo được người dùng hay các khách hàng vào bình luận trong bài viết về sản phẩm của mình nhiều hơn? Làm sao giúp sản phẩm của mình được nhiều người biết tới hơn?
- Nhà sản xuất làm sao cần biết phản hồi của dùng với sản phẩm của mình? Để từ đó có thể cải thiện về những sản phẩm sắp tới?

Một số phương pháp trong nghiên cứu này đã giải quyết các vấn đề trên, nhưng trong tương lai sẽ cần nhiều nghiên cứu hơn để giải quyết triệt để những thách thức này.

1.5. Phương pháp nghiên cứu

Phương pháp thu thập thông tin: Thu thập thông tin dạng văn bản từ các trang tin tức uy tín, hàng đầu và phù hợp với các tiêu chí của đề tài cũng như nằm trong lĩnh vực nghiên cứu.

Phương pháp phân tích và tổng hợp lý thuyết: Nghiên cứu những tài liệu, sách và bài báo khoa học liên quan để tiếp thu những mô hình, kiến thức, công nghệ mới phục vụ cho quá trình nghiên cứu..

Phương pháp thống kê: Thống kê những đoạn văn bản, những từ ngữ để hiểu dữ liệu cũng như trích xuất những đặc điểm quan trọng phù hợp cho việc máy tính dự đoán sau này.

Phương pháp định lượng: Chuyển những đoạn văn bản thành những số liệu thống kê có ý nghĩa, áp dụng vào những mô hình toán học cũng như thuật toán để đạt được kết quả dự đoán đúng với mục đích đề tài.

1.6. Mục tiêu nghiên cứu

Đề tài nghiên cứu khoa học "Khai phá ý kiến người dùng" sẽ bổ trợ những kiến thức trong lĩnh vực khai phá dữ liệu và khoa học máy tính cho các sinh viên, cung cấp tài liệu nghiên cứu về lĩnh vực này cho các công trình khác. Đem lại những định hướng phát triển trong phạm vi quy mô nhất định, đưa ra những ý tưởng phát triển khác trong lĩnh vực.

Nghiên cứu xuất phát từ nhu cầu của các doanh nghiệp, sử dụng các thuật toán tối ưu nhất hiện tại để đánh giá và nghiên cứu cũng như khai phá ý kiến người dùng trong lĩnh vực này. Từ đánh giá, nhận xét, phản hồi của người dùng, chúng tôi xây dựng giải pháp và rút trích thông tin thiết yếu cho doanh nghiệp cải thiện sau này.

1.7. Đối tượng và phạm vi nghiên cứu

Phạm vi đề tài tập trung trong lĩnh vực điện thoại thông minh phân tích nội dung các bài tin tức đưa tin về sản phẩm sắp ra mắt, ứng dụng vào các thuật toán máy học tối ưu để dự đoán và trích xuất ý kiến, cảm xúc của người dùng nhằm khai thác ý kiến một cách chính xác nhất để có thể biết được nhu cầu cũng như sự mong đợi của khách hàng đối với sản phẩm.. Phạm vi không gian là những bài đưa tin về sản phẩm và những sự kiện xoay quanh sản phẩm đó cũng như các bình luận của người dùng, điều này thể hiện độ hứng thú cũng như tích cực/tiêu cực đối với sản

phẩm. Phạm vi giới hạn nội dung chỉ nghiên cứu trong lĩnh vực điện thoại thông minh và những bài đưa tin, bình luận (ở dạng văn bản) ở báo điện tử chính thống và không xét đến những yếu tố khác.

Ngoài ra, đề tài còn xem xét, tối ưu thuật toán thu thập và tiền xử lý dữ liệu văn bản để phù hợp với ngôn ngữ, thị trường cũng như văn hóa Việt Nam, điều này sẽ mở ra nhiều ý tưởng và hướng nghiên cứu về NLP (Natural Language Processing – Xử lý ngôn ngữ tự nhiên) nói riêng và khai phá dữ liệu nói chung trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Khai phá văn bản

Khai phá văn bản là quá trình khám phá và phân tích một lượng lớn dữ liệu văn bản phi cấu trúc được hỗ trợ bởi phần mềm có thể xác định các khái niệm, dữ liệu con, chủ đề, từ khóa và các thuộc tính khác trong dữ liệu.

2.2. Công cụ

2.2.1. Jupyter python

Jupyter được xây dựng bởi Project Jupyter, là một tổ chức phi lợi nhuận được thành lập nhằm mục đích "xây dựng phần mềm mã nguồn mở, tiêu chuẩn mở và dịch vụ cho tính toán tương tác (interactive computing) giữa hàng loạt các ngôn ngữ lập trình". Phát triển từ IPython do Fernando Pérez tạo từ 2014, Project Jupyter hỗ trợ môi trường thực thi rất nhiều ngôn ngữ khác nhau. Tên tổ chức bắt nguồn từ ba ngôn ngữ lập trình chính mà Jupyter hỗ trợ, là Julia, Python và R.

Jupyter Notebook (IPython Notebooks) là một môi trường tính toán tương tác dựa trên web để tạo các notebook. "Notebook" ở đây có thể hiểu theo nhiều cách, thường là ứng dụng web, máy chủ hoặc văn bản. Văn bản được lưu dưới dạng JSON, có lược đồ, danh sách ô đầu vào và đầu ra, ghi chú, phép toán, sơ đồ và đa phương tiện, đuôi mở rộng thường là ".ipynb". Sản phẩm từ Jupyter Notebook có thể chuyển sang nhiều hình thức chuẩn khác như HTML, slide trình chiếu, LaTeX, PDF, ReStructuredText, Markdown, Python) bằng cách thực thi dòng lệnh có sử dụng thư viện nbconvert trên web. Để cho phép sử dụng nhiều ngôn ngữ lập trình, Jupyter Notebook kết nối được với rất nhiều kernel. Jupyter kernel là một chương trình chịu trách nhiệm xử lý nhiều loại yêu cầu (thực thi

code, hoàn thành code, gỡ lỗi lập trình) từ người dùng và trả về kết quả. Kernel tương tác với các tính năng khác của Jupyter thông qua ZeroMQ trên mạng lưới và có thể kết nối với nhiều khách hàng cùng một lúc. IPython là kernel mặc định và tham chiếu qua ipykernel wrapper. Đối với các ngôn ngữ khác nhau thì kernel có chất lượng và tính năng khác nhau.

2.2.1.1. Terminal IPython

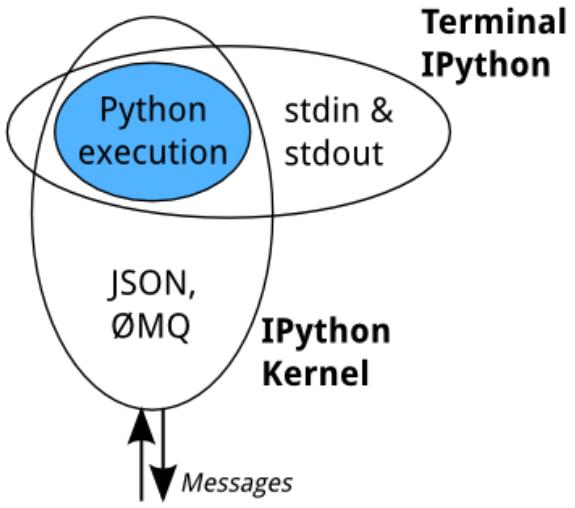
Khi nhập ipython, ta có giao diện IPython mặc định ở terminal và nó thực thi tương tự như sau:

```
while True:
    code = input(">>> ")
    exec(code)
```

Tất nhiên bản chất phức tạp hơn nhiều nhưng mô hình thì tương tự như trên: yêu cầu người dùng nhập code và thực thi theo đúng quy trình. Mô hình này thường được gọi là REPL (Read-Eval-Print-Loop).

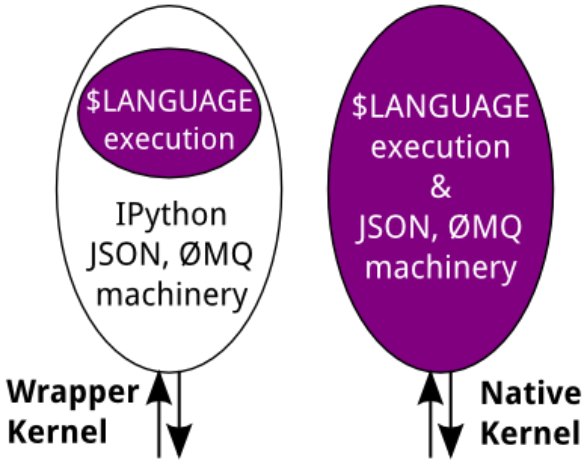
2.2.1.2. Ipython Kernel

Tất cả các giao diện khác như Notebook, At console, ipython console trong terminal và giao diện từ bên thứ ba đều sử dụng IPython Kernel. Đây là quy trình tách rời chịu trách nhiệm chạy code của người dùng và thực thi một số tính toán. Front-ends tương tác với IPython Kernel bằng cách gửi tin nhắn JSON qua trung gian là ZeroMQ. Bộ máy thực thi lõi cho kernel được chia sẻ với terminal IPython:



Hình 2-1. IPython Kernel

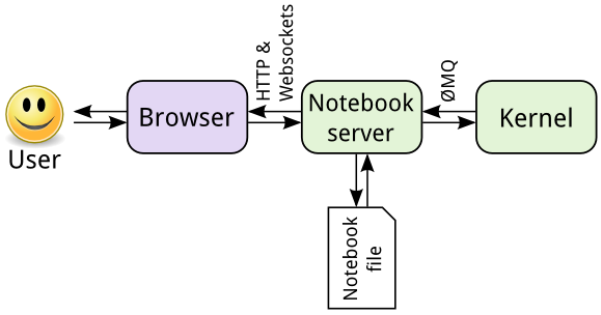
Một quy trình kernel có thể kết nối nhiều frontend cùng lúc. Trong trường hợp này, các frontend khác nhau sẽ truy cập các biến giống nhau. Cách thiết kế này nhằm mục đích cho phép việc phát triển nhiều frontend dễ dàng hơn vì sử dụng cùng một kernel, đồng thời có thể hỗ trợ ngôn ngữ mới trong cùng frontend. Hiện nay có hai cách để phát triển kernel cho ngôn ngữ mới: wrapper kernel và native kernel.



Hình 2-2. Wrapper Kernel và Native Kernel

2.2.1.3. Notebook

Giao diện Notebook thực hiện nhiều tính năng hơn là kernel. Bên cạnh việc thực thi code thì Notebook cũng lưu trữ code và kết quả cùng với các ghi chú trong không gian chỉnh sửa được. Khi lưu, tài liệu được gửi từ trình duyệt web đến máy chủ và được lưu vào ổ đĩa dưới dạng file JSON với đuôi mở rộng là .ipynb.

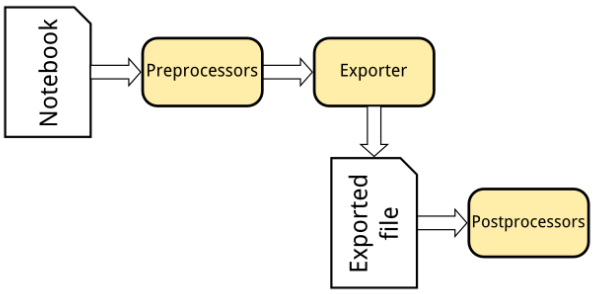


Hình 2-3. Notebook

Máy chủ của notebook chịu trách nhiệm lưu và tải notebook để người dùng có thể chỉnh sửa ngay cả khi không có kernel cho ngôn ngữ đang sử dụng, nhưng không thực thi được code. Kernel không biết bất cứ thứ gì liên quan đến notebook mà chỉ nhận code để thực thi mỗi lần người dùng chạy code.

2.2.1.4. Xuất file

Công cụ Nbconvert cho phép Jupyter chuyển notebook sang các định dạng khác thông qua các bước sau:



Hình 2-4. Xuất file notebook

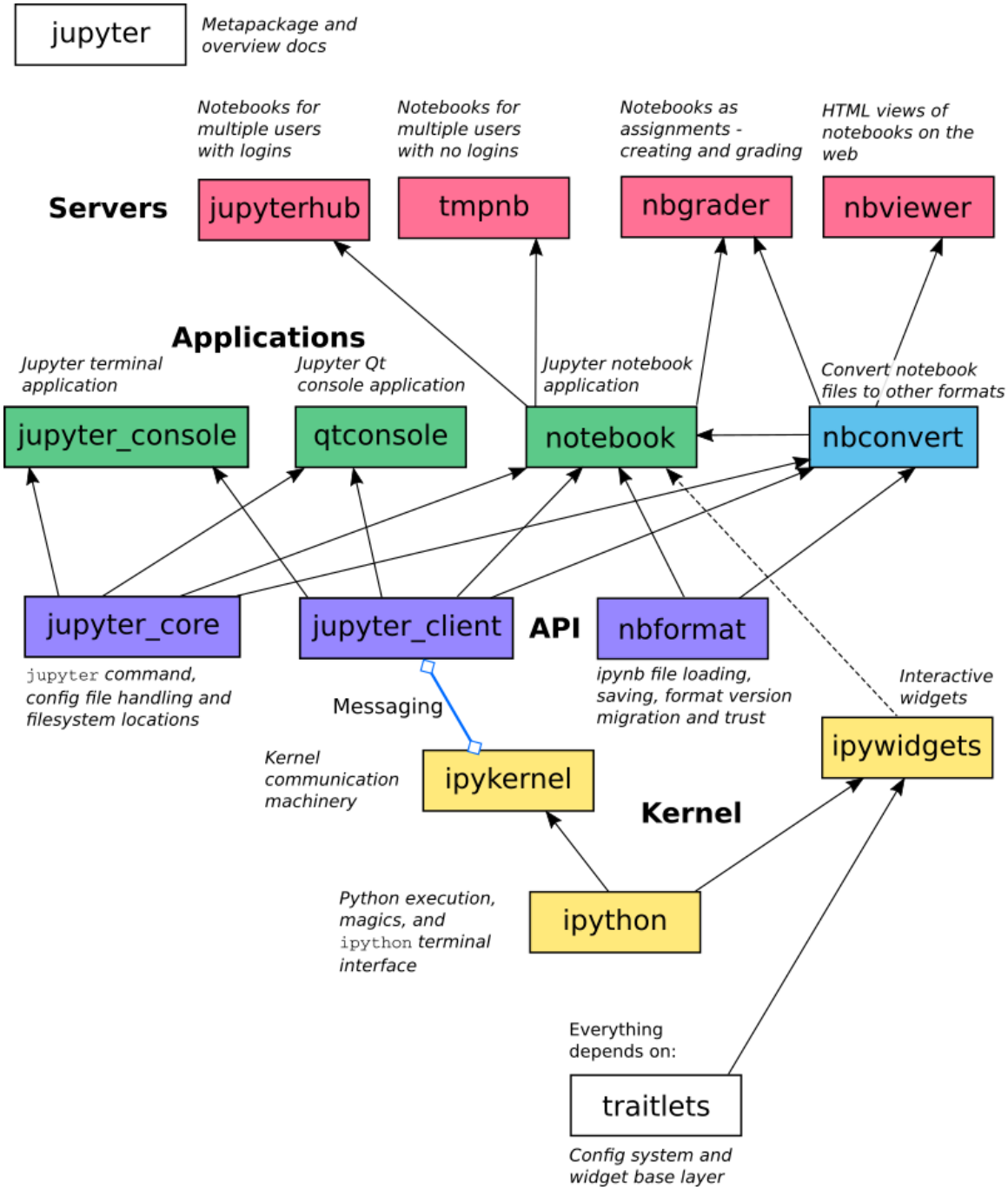
Đầu tiên, tiền xử lý chỉnh sửa notebook trong ô nhớ. Sau đó tính năng chuyển đổi sẽ tạo file theo định dạng được yêu cầu. Cuối cùng, hậu xử lý sẽ giải quyết vấn đề về file được chuyển đổi.

Trang web nbviewer sử dụng nbconvert với HTML exporter. Khi được cung cấp một URL, exporter sẽ nhận notebook từ URL đó, chuyển thành HTML và cung cấp HTML đó cho người dùng.

2.2.1.5. IPython.parallel

IPython cũng cung cấp nền tảng tính toán song song, IPython.pool. Công cụ này là phiên bản mở rộng của IPython được mô tả ở trên, cho phép người dùng điều khiển nhiều công cụ riêng lẻ cùng lúc.

2.2.1.6. Tổng quan dự án



Hình 2-5. Tổng quan dự án Jupyter

2.2.2. Pycharm

2.2.2.1. Tổng quan

PyCharm là một môi trường phát triển tích hợp (IDE) được sử dụng cho lập trình máy tính, đặc biệt là lập trình bằng ngôn ngữ Python. Phần mềm này được xây dựng bởi công ty JetBrains ở Cộng hòa Séc. Pycharm cung cấp tính năng phân tích code, trình gỡ lỗi đồ họa, trình kiểm tra đơn vị tích hợp, tích hợp với các hệ thống kiểm soát phiên bản (VCSes) và hỗ trợ phát triển web với Django cũng như Khoa học dữ liệu với Anaconda. PyCharm là công cụ đa nền tảng, có các phiên bản Windows, macOS và Linux.

2.2.2.2. Tính năng

- Hỗ trợ và phân tích code
- Điều hướng dự án
- Tái cấu trúc Python
- Hỗ trợ cho các khung web
- Trình gỡ lỗi Python tích hợp
- Kiểm thử đơn vị tích hợp
- Phát triển Google App Engine Python
- Tích hợp kiểm soát phiên bản
- Hỗ trợ cho các công cụ khoa học

2.2.2.3. Plugins

PyCharm cung cấp API để lập trình viên có thể tự viết plugin mở rộng tính năng PyCharm. Một số plugin từ JetBrains IDE khác cũng hoạt động với PyCharm. Có hơn 1000 plugin tương thích với PyCharm.

2.2.2.4. Ứng dụng

PyCharm, được cho là "bùa mê" các nhà phát triển Python ở mọi cấp độ, được sử dụng trong giai đoạn đầu tiên của nghiên cứu là bước cào dữ liệu. Chúng tôi lựa chọn PyCharm vì đây là một nền tảng khá phổ biến và có nhiều tính năng thông minh như bộ code completion, dễ dàng điều hướng và kiểm tra lỗi; IDE này có thể tự động thụt lề, phát hiện văn bản trùng lặp và kiểm tra lỗi. Ngoài ra PyCharm có các tính năng tìm kiếm mã nguồn thông minh để tìm kiếm từng từ một gần như ngay lập tức, có chế độ Go-To, chế độ Lens để di chuột tới gần code và highlight nó để quay lại làm việc sau.

2.2.3. Google Colab

2.2.3.1. Tổng quan

Colaboratory, viết tắt là "Colab", cũng là nền tảng để viết và thực thi lệnh Python, nhưng thực hiện ngay trên trình duyệt web. Sử dụng Colab thì không cần cấu hình, truy cập được GPU miễn phí và dễ dàng chia sẻ.

Colab notebook cho phép ghi chép cả code và văn bản trong một file duy nhất, có thể chèn ảnh, HTML, LaTeX và một số định dạng khác. Các notebook Colab được lưu vào drive của người dùng nên dễ dàng chia sẻ cho đồng nghiệp, bạn bè và thực hiện được các chức năng như bình luận hoặc chỉnh sửa. Thật ra colab bản chất cũng là Jupyter notebook, nhưng máy chủ là của Colab.

Colab là một công cụ rất phù hợp cho lĩnh vực khoa học dữ liệu hay máy học. Chủ yếu là vì Colab tối ưu hóa các thư viện Python bằng cách phân tích và trực quan hóa dữ liệu. Người dùng cũng có thể thêm dữ liệu từ Google Drive, bao gồm google sheets, từ Github và nhiều nguồn khác. Ngoài ra, lập trình viên có thể nhập dữ liệu là hình ảnh, phân loại trên hình ảnh hay đánh giá mô hình. Colab notebook thực thi trên máy chủ đám mây của google nên tận dụng được

công nghệ phần cứng của Google, bao gồm GPU và TPU, chứ không bị phụ thuộc vào tốc độ của máy tính cá nhân.

Một số ứng dụng về máy học có thể kể đến của Colab là: Bắt đầu với TensorFlow, Phát triển và đào tạo cấu trúc mạng, phử nghiệm với TPU, phổ biến nghiên cứu AI, viết hướng dẫn,...

2.2.3.2. Ứng dụng

Google Colab là nền tảng được sử dụng gần như xuyên suốt quá trình nghiên cứu, đặc biệt là trong bước xây dựng và thực thi các mô hình như SVM, LSTM, BiLSTM, BERT.

2.2.4. Chromium

Đây là trình duyệt web mã nguồn mở mà chúng tôi chọn để làm công cụ giao tiếp giữa Python và các trang đích chứa dữ liệu cần phân tích.

2.3. Thư viện

Dựa trên lập trình bằng ngôn ngữ Python, chúng tôi sử dụng một số thư viện sẵn có để giúp cho việc khai phá dữ liệu, các thư viện đó lần lượt được liệt kê dưới đây:

2.3.1. Request:

Đây là thư viện giúp người dùng có thể gửi các request HTTP/1.1 đơn giản nhất. Thư viện xử lý để giúp người dùng không phải thêm các chuỗi truy vấn (query strings) một các thủ công vào các URLs, form-encode hoặc PUT và POST dữ liệu, thư viện hoạt động tốt với JSON Method.

2.3.2. Beautiful Soup

Beautiful Soup là một thư viện giúp bạn dễ dàng lấy thông tin từ các trang web. Nó nằm trên một trình phân tích cú pháp HTML hoặc XML, cung cấp các thành ngữ thuần Python (Pythonic) để lặp lại, tìm kiếm và sửa đổi cây phân tích cú pháp.

2.3.3. Pandas

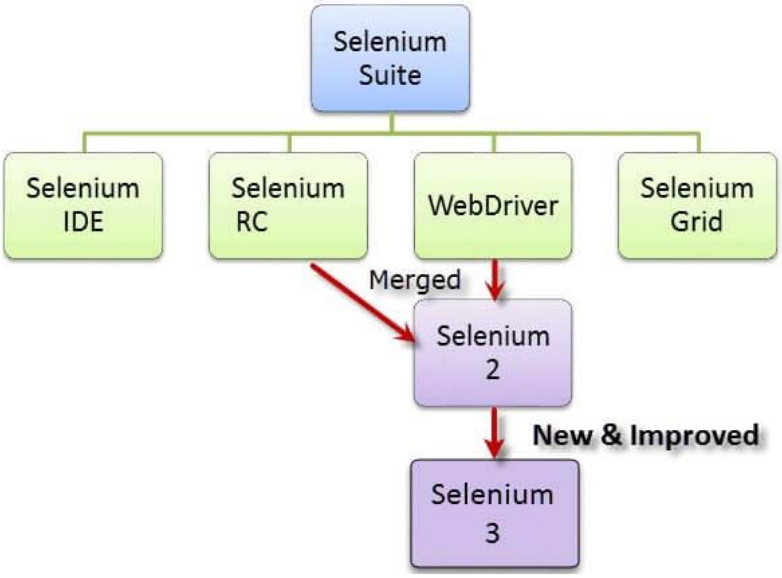
Pandas là một gói thư viện viết bằng Python phổ biến cho khoa học dữ liệu và với lý do như: nó cung cấp các cấu trúc dữ liệu mạnh mẽ, linh hoạt giúp các thao tác và phân tích dữ liệu dễ dàng hơn. DataFrame là một trong những cấu trúc dữ liệu rất mạnh của Pandas. Pandas kết hợp các tính năng tính toán mảng hiệu suất cao của NumPy với khả năng thao tác dữ liệu linh hoạt của bảng tính và cơ sở dữ liệu quan hệ (như SQL). Nó cung cấp chức năng lập chỉ mục chính xác để giúp dễ dàng định hình lại, cắt và trộn, thực hiện tổng hợp và chọn tập hợp dữ liệu. Pandas là công cụ chính mà chúng ta sẽ sử dụng trong bài báo này để xử lý dữ liệu.

	index	comment	label
0	0	Cá nhân mình thấy cam của lenovo trước đây chụ...	2
1	1	Lenovo cứ chạy đua thiết kế mà chất lượng chẳn...	2
2	2	Khi nào click hoạt camera trước thì nó tự động...	2
3	3	Chết cười với so sánh độc đáo của bạn)	2
4	4	Chẳng bít là sp j nhưng đặt tên là z5 thì chắc...	1
5	5	Bác thật thâm thúy	0
6	6	Vấn đề là họ đã ăn đứt cái tai thỏ nhà bạn, ch...	2
7	7	Không biết SS ra tai thỏ hoặc AP trang bị vân ...	0
8	8	Samurai đã lộ nguyên hình	0
9	9	hình vẽ y hệt iphone X thiếu tai thỏ	2

Hình 2-6. Cấu trúc DataFrame của dữ liệu Bình luận trong bài báo

2.3.4. Selenium

Không chỉ là một library mà là một framework mã nguồn mở, miễn phí giúp kiểm tra tự động các ứng dụng được phát triển trên các trình duyệt, nền tảng khác nhau. Bạn có thể sử dụng nhiều ngôn ngữ lập trình như Java, C #, Python, v.v để tạo tập lệnh kiểm tra Selenium. Selenium sinh ra với mục đích ban đầu là kiểm thử phần mềm, tuy nhiên trong nghiên cứu này, chúng tôi sử dụng Selenium để tự động hóa các thao tác trên trình duyệt phục vụ việc thu thập dữ liệu. Selenium không chỉ là một công cụ đơn lẻ mà là một bộ phần mềm, mỗi phần phục vụ cho các nhu cầu thử nghiệm khác nhau của một tổ dự án. Dưới đây là mô tả cho cấu trúc của Framework Selenium.



Hình 2-7. Cấu trúc tổ chức của Framework Selenium

Trong phạm vi bài nghiên cứu này, chúng tôi kết hợp sử dụng Selenium với ChromeDriver để thực thi Selenium Web Driver nhằm mục đích điều khiển trình duyệt Chrome. Web Driver cung cấp các khả năng để điều hướng đến các trang web, đầu vào của người dùng, thực thi các đoạn mã JavaScript và nhiều ứng dụng khác. ChromeDriver là một máy chủ độc lập thực hiện tiêu chuẩn W3C WebDriver. ChromeDriver có sẵn cho Chrome trên Android và Chrome trên Máy tính để bàn (Mac, Linux, Windows và ChromeOS).

2.3.5. Tensorflow

TensorFlow là một thư viện phần mềm mã nguồn mở, được dùng ở lĩnh vực máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. Hiện tại ngôn ngữ này được sử dụng cho rất nhiều sản phẩm cả nghiên cứu và thương mại của Google, ví dụ như nhận dạng giọng nói, Gmail, Google Photos, và bộ máy tìm kiếm. Ban đầu TensorFlow được phát triển bởi đội Google Brain nhằm phục vụ mục đích nghiên cứu và sản xuất của Google, sau đó vào ngày 9/11/2015 được phát hành theo giấy phép mã nguồn mở Apache 2.0.

2.3.5.1. Tính năng

TensorFlow cung cấp API Python và C, API không đảm bảo khả năng tương thích ngược: C++, Go, Java, JavaScript, Swift. Tính năng đóng gói từ bên thứ ba được sử dụng cho C#, Haskell, Julia, MATLAB, R, Scala, Rust, OCaml,and Crystal. Ngôn ngữ mới nên được xây dựng trên C API. Tuy nhiên, không phải tất cả tính năng đều đã có sẵn, nên người dùng có thể kết hợp sử dụng Python API.

2.3.5.2. Ưu điểm

Thứ nhất, người dùng có thể dễ xây dựng mô hình khi sử dụng TensorFlow vì TensorFlow cung cấp nhiều tầng tính trừu tượng để người dùng tùy chọn. Ví dụ như sử dụng API Keras bậc cao sẽ phù hợp và dễ dàng cho người mới sử dụng. Nếu cần độ linh hoạt hơn, phần mềm có cung cấp trình gỡ lỗi trực quan và tương tác được. Đối với các dự án lớn về máy học, có thể sử dụng Distribution Strategy API để phân chia tác vụ trên nhiều cấu hình phần cứng khác nhau mà không cần thay đổi định nghĩa mô hình.

Thứ hai, người dùng có thể sử dụng xây dựng máy học ở bất kỳ đâu. TensorFlow luôn có đường dẫn trực tiếp đến nơi người dùng muốn sử dụng. Có thể là trên máy chủ, các thiết bị hay trên web, dù là ngôn ngữ hoặc nền tảng nào cũng đều có thể kết nối TensorFlow. Chỉ cần người dùng linh hoạt trong chọn lựa sản phẩm của TensorFlow: sử dụng TensorFLow mở rộng, TensorFlow Lite, TensorFlow.js.

Thứ ba, dễ dàng thực hiện thí nghiệm nghiên cứu mà không phải lo lắng về tốc độ hoặc hiệu suất. Người dùng linh hoạt trong kiểm soát những vấn đề phức tạp về Tôpô nhờ vào tính năng như Keras Functional API và Model Subclassing API. TensorFlow cũng có một hệ sinh thái về thư viện và mô hình

để thí nghiệm như Ragged Tensors, TensorFlow Probability, Tensor2Tensor và BERT.

2.3.5.3. Ứng dụng

Tensorflow được xuất hiện trong nghiệp vụ xây dựng mô hình, đặc biệt là ở mô hình LSTM và BiLSTM.

2.3.6. Sklearn

2.3.6.1. Tổng quan

Scikit-learn (trước là scikits.learn, còn được gọi là sklearn) là một thư viện phần mềm máy học miễn phí dành cho ngôn ngữ lập trình Python. Các tính năng của thư viện áp dụng cho thuật toán phân lớp, đệ quy, gom cụm, bao gồm support vector machines, random forests, gradient boosting, k-means và DBSCAN, đồng thời thư viện này được thiết kế để phối hợp với thư viện số Python và các thư viện cụ thể như NumPy và SciPy.

Hình 2-8. Scikit.learn

Scikit-learn là một trong những thư viện máy học nổi tiếng nhất trên cộng đồng GitHub, phần lớn được viết bằng Python, và một số thuật toán viết bằng python để tăng hiệu suất. Support vector machines được thực thi từ LIBSVM; logistic regression và linear support vector machines từ LIBLINEAR. Trong những trường hợp này thì người dùng không thể mở rộng phương pháp với bằng Python. Scikit-learn phối hợp tốt với rất nhiều thư viện

Python khác như matplotlib và plotly để đánh dấu, đánh số cho các vector mảng, pandas dataframes, scipy, ...

2.3.6.2. Ứng dụng

Sklearn được sử dụng để phân loại Support Vector Machine để huấn luyện và đánh giá mô hình.

2.3.7. Keras

2.3.7.1. Tổng quan

Keras là một thư viện cấu trúc mạng mã nguồn mở được viết bằng Python. Thư viện này có thể chạy dựa trên TensorFlow, Microsoft Cognitive Toolkit, R, Theano, hoặc PlaidML. Thư viện này được thiết kế để thực hiện các thí nghiệm nhanh với neural networks, tập trung vào tính thân thiện với người dùng, mô-đun hóa và có thể mở rộng. Thư viện được phát triển như một phần dự án nghiên cứu ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), mà tác giả chính cũng là người duy trì nó là François Chollet, một kỹ sư tại Google. Chollet cũng là tác giả của mô hình deep neural network.

Vào năm 2017, đội ngũ TensorFlow của Google quyết định hỗ trợ Keras trong thư viện chính của TensorFlow. Chollet giải thích rằng anh mong muốn Keras là một giao diện hơn chỉ là một nền tảng máy học đơn thuần. Keras cung cấp tính trừu tượng bậc cao và trực quan để người dùng dễ dàng phát triển các mô hình học sâu dù là sử dụng trên môi trường backend nào. Microsoft cũng đã bổ sung CNTK backend cho Keras trong CNTK phiên bản 2.0.

2.3.7.2. Tính năng

Keras bao gồm rất nhiều triển khai của các khối cấu trúc mạng để làm việc với dữ liệu hình ảnh và văn bản dễ dàng hơn, đơn giản hóa code cần thiết

để viết code cho cấu trúc mạng sâu. Code được lưu trữ trên GitHub và các diễn đàn hỗ trợ cộng đồng như GitHub và Slack.

Bên cạnh các cấu trúc mạng chuẩn, Keras còn hỗ trợ cho cấu trúc mạng tích chập, cấu trúc mạng hồi quy. Ngoài ra còn hỗ trợ các ứng dụng phổ biến khác như dropout (bỏ học), batch normalization, và pooling.

2.3.7.3. Ứng dụng

Chúng tôi sử dụng thư viện Keras trong thao tác trích xuất đặc trưng, cụ thể là trong phương pháp TF-IDF. Thao tác này được thực hiện sau bước tiền xử lý và trước khi training model. Ngoài ra, cùng với sklearn và bert thì thư viện keras cũng là một thành phần quan trọng trong việc xây dựng các mô hình.

2.4. Các nghiên cứu trước đó

Các nghiên cứu trước đây [3] [4] [7] đã giới thiệu các phương pháp phân tích dữ liệu phi cấu trúc, đặc biệt là dữ liệu văn bản theo cách thức truyền thống như xác định stopwords, Part of Speech (POS), Túi từ (Bag of Word – BoW), TF-IDF cũng như sử dụng mô hình SVM [25] nhằm phân loại văn bản. Minqing Hu & Bing Liu [20] đã sử dụng các phương pháp truyền thống này để tóm tắt các đánh giá của khách hàng và John Brandt [15] đã áp dụng chúng trong phân loại tài liệu chính trị.

Đi cùng với sự phát triển thần tốc của Deep Learning (Học sâu), trên thế giới liên tiếp xuất hiện nhiều mô hình dựa trên công nghệ này được sử dụng nhằm giải quyết các nhiệm vụ xử lý ngôn ngữ tự nhiên hiệu quả hơn. Thuật toán Long-short term memory (LSTM) [22] là một trong số đó. Thuật toán này dựa trên các từ trước đó cùng ngữ cảnh để dự đoán các từ phía sau và cũng được sử dụng trong phân loại văn bản, cải thiện độ chính xác của mô hình một cách đáng kể.

Gần đây, một số công trình nghiên cứu khoa học đã chứng minh rằng các mô hình được huấn luyện trước trên dữ liệu văn bản lớn (pre-trained model) đã đem lại kết quả tối ưu hơn trong phân loại văn bản và các nhiệm vụ NLP khác. Một trong những mô hình này là word-embeddings (nhúng từ) như word2vec [23] và Glove [13] hoặc word-embeddings theo ngữ cảnh, như CoVe [4] và Elmo [18]. Các phương pháp này được dùng để thêm các đặc trưng bổ sung nhằm tối ưu hóa kết quả. Một mô hình huấn luyện trước khác cũng đem lại kết quả đáng kể là sentence-level (mô hình ở mức độ câu chữ). Howard và Ruder [14] đã đề xuất ULMFit, một mô hình được huấn luyện trước với sự tinh chỉnh (fine-tuning) đã đạt được kết quả ấn tượng với sáu bộ dữ liệu văn bản lớn. Ngoài ra cũng có một số mô hình huấn luyện trước cũng có hiệu suất ấn tượng trong việc xử lý một lượng lớn dữ liệu chưa được gắn nhãn như OpenAI GPT [1] và BERT [12].

Cấu trúc mô hình BERT là một bộ mã hóa Transformer hai chiều dựa trên mô hình Transformer ban đầu [2] được sử dụng cho một số tác vụ NLP, đặc biệt là phân loại văn bản.

Trên thế giới đã có nhiều nghiên cứu áp dụng BERT với phân loại văn bản trong thực tế. Mahsa Shafaei, Niloofar Safi Samghabadi, Sudipta Kar và Thamar Solorio [16] đã đánh giá bộ phim dựa trên đánh giá của người dùng và kịch bản, lời thoại của bộ phim đó. Manish Munikar, Sushil Shakya và Aakash Shrestha [17] dự đoán những đánh giá cảm xúc của người dùng thông qua bình luận của họ. Bên cạnh đó, một số nghiên cứu [10] [19] [21] cũng đạt được kết quả đáng chú ý trong nhiệm vụ phân loại tài liệu với văn bản dài, phức tạp theo ngữ cảnh. Hơn nữa, các nghiên cứu khác [6] [11] [27] giới thiệu các phương pháp tinh chỉnh BERT và tối ưu hóa mô hình này trong nhiệm vụ phân loại văn bản.

Tuy nhiên, hiện tại không có thuật toán nào có thể phân tích tin tức, đánh giá người dùng và dự đoán ảnh hưởng của chúng với hành vi khách hàng, đặc biệt được

áp dụng trong tin tức Việt Nam với các mô hình huấn luyện trước này. Nghiên cứu này không chỉ phân tích và dự đoán về ngôn ngữ tiếng Việt, mà còn kết hợp hai mô hình nhằm đánh giá đa khía cạnh và hiệu quả cả nội dung bản tin và ý kiến của người dùng đối với suy nghĩ và hành vi của khách hàng tiềm năng.

CHƯƠNG 3. PHƯƠNG PHÁP

3.1. Thu thập dữ liệu và dán nhãn

Trước tiên, để lý giải cho lý do trong phạm vi nghiên cứu này, chúng tôi chọn trang VnExpress.net để thu thập dữ liệu nhằm đánh giá các mô hình dự đoán vì các lý do sau như: hiện tại theo báo cáo của SimilarWeb, VnExpress đang là trang tin thức có số lượng truy cập và độc giả lớn nhất Việt Nam, với tổng lượng truy cập vào khoảng 137 triệu lượt, lớn hơn rất nhiều so với đối thủ thứ hai là trang 24h.com.vn với chỉ 91 triệu lượt. Bên cạnh đó, khi so sánh về độ đa dạng của độc giả, chúng tôi nhận thấy VnExpress là một trong những trang có lượng độc giả đa dạng nhất, với khoảng 13% lượng truy cập đến từ nước ngoài khi so sánh với các trang như , zingnews.vn (7%), dantri.vn (5%), 24h.com.vn (3%) … Xét về mức độ chuyên môn, theo khảo sát của chúng tôi, hiện tại VnExpress là một trang thuộc công ty FPT Online, một nhánh của tập đoàn FPT - vốn dĩ là một công ty công nghệ. Việc là tờ báo của một công ty công nghệ phần nào sẽ giúp cho đội ngũ biên tập của VnExpress có kiến thức rộng hơn về mảng điện tử, mà cụ thể là điện thoại thông minh khi so với một số tờ báo khác. Đó là những lý do mà chúng tôi quyết định tập trung nguồn lực để khai thác thông tin từ trang này, mặc dù sẽ có những rủi ro về tính khách quan của thông tin thu thập được, nhưng nhóm chúng tôi tin rằng những sai số đó là không đáng kể dựa trên những luận điểm đã trình bày ở trên .

Về cách thức thực hiện, chúng tôi sẽ mô tả quy trình gồm hai quy trình:

3.1.1. Thu thập dữ liệu:

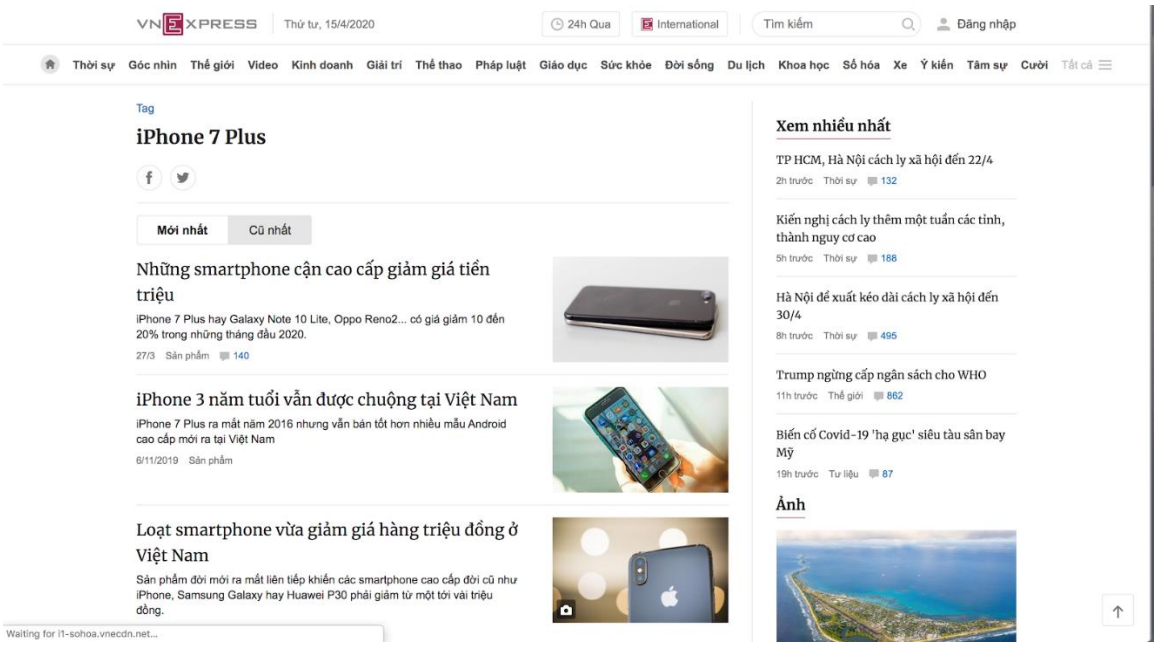
3.1.1.1. Thư viện sử dụng:

- Request
- BeautifulSoup

- Pandas
- Regular expression (re),
- Selenium,
- Time

3.1.1.2. Quá trình thu thập dữ liệu

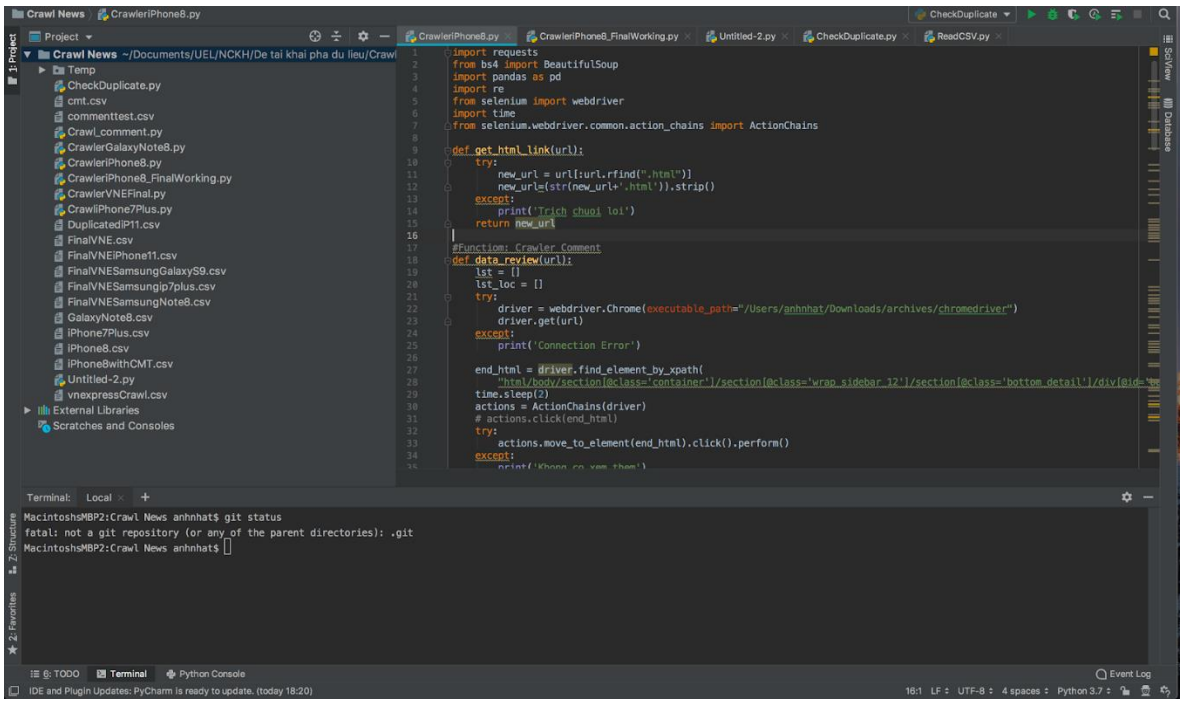
Nhóm nghiên cứu đã thực hiện việc lấy các mẫu dữ liệu trên về từ sản phẩm trên trang VnExpress.net dựa trên các tag sản phẩm có sẵn của báo:



Hình 3-1. Dữ liệu thô trên trang VnExpress.net

Để lấy được các bình luận và nội dung bài viết dưới dạng text văn bản, thu được một lượng mẫu đủ lớn cho báo cáo, nhóm nghiên cứu đã dựa trên các liên kết và bình luận được thu thập thủ công trước đó. Sau khi thu thập các link là các bài viết có chứa bình luận là dữ liệu mẫu, nhóm đưa vào phần mềm để thu thập tự động tất cả các bình luận, cụ thể:

- Sử dụng thư viện Request để gửi một HTTP Request lên server của VnExpress.net để lấy một chuỗi các link bài báo về các sản phẩm liên quan.
- Tiếp tục từ tập hợp link này, liên tục gửi request với độ trễ bất biến lên VnExpress.net để lấy dữ liệu HTML của bài viết về thông qua thư viện request.
- Sử dụng thư viện request và selenium để lấy dữ liệu bình luận về, do tính chất phân trang bình luận của VnExpress, nên ta dùng thư viện selenium đóng vai trò tự động hóa quá trình cuộn trang.
- Dùng thư viện Beautiful Soup kết hợp trong quá trình lấy dữ liệu để lọc ra các nội dung trong file HTML cần thiết, lưu tất cả dữ liệu vào DataFrame thông qua thư viện pandas.



Hình 3-2. Dữ liệu được thu thập tự động bằng phần mềm

Cuối cùng, sau khi khai phá dữ liệu, nhóm đã khai thác được hơn 10000 dòng dữ liệu về bình luận, hơn 5000 dòng về bài báo, trải qua các quá trình xử

lý phía sau, nhóm đã trích lọc được 2500 dòng bài báo mẫu và hơn 8700 dòng bình luận mẫu để đưa vào quá trình kiểm thử các mô hình.

3.1.1.3. Môi trường thực nghiệm

Hệ điều hành	Windows 10 Education
Vi xử lý	2.70 GHz
Ram	8.00 GB
Tốc độ mạng trung bình	50.0 Mbps
Ngôn ngữ khai phá dữ liệu	Python
Miền dữ liệu	Bài viết về đánh giá sản phẩm, tổng hợp thông tin sản phẩm, bình luận của độc giả về sản phẩm điện thoại thông minh.
Nguồn dữ liệu	VnExpress.net
Tổng số bình luận thu thập và trích lọc được	8700
Tổng số bài viết thu thập và trích lọc được	2500

Bảng 3-1. Mô tả môi trường thực nghiệm

3.1.2. Dán nhãn dữ liệu:

Sử dụng nền tảng chính là Pycharm và các thư viện hỗ trợ như BeautifulSoup, Selenium, Pandas,… chúng tôi thực hiện lấy dữ liệu tự động hàng loạt các bài báo của VnExpress.net. Tuy nhiên nhằm phân loại theo đúng sản

phẩm, chúng tôi phải sử dụng bộ máy tìm kiếm để lọc theo từ khóa bao gồm tên thương hiệu và tên mẫu mã chính xác, sau đó mới áp dụng kỹ thuật thu thập dữ liệu cho kết quả vừa tìm được.

Tập dữ liệu của chúng tôi bao gồm hai phần: nội dung tin tức và bình luận trong tin tức được thu thập duy nhất từ trang VnExpress.net. Các dữ liệu này được dán nhãn với ký hiệu như sau:

Nhãn	**Ý nghĩa**	**Tiêu chí gắn nhãn chính**
0	Dữ liệu mang tính chất trung tính (neutral).	Chủ đề bình luận (bài viết) hướng tới nằm ngoài phạm vi bài viết (sản phẩm), không liên quan, không ảnh hưởng.
1	Dữ liệu mang tính chất tích cực (positive).	Chủ đề bình luận (bài viết) hướng tới ảnh hưởng tích cực tới chủ đề bài viết (sản phẩm), tạo sự hứng thú, kích thích người đọc.
2	Dữ liệu mang tính chất tiêu cực (negative).	Chủ đề bình luận (bài viết) hướng tới ảnh hưởng tiêu cực tới chủ đề bài viết (sản phẩm), tạo cảm giác không tốt, giảm ham muốn sử dụng sản phẩm..

Bảng 3-2. Nhãn dữ liệu

Thao tác dán nhãn nhằm mục đích phục vụ cho giai đoạn nghiên cứu phía sau và được thực hiện bởi nhóm người trong độ tuổi 18-24. Theo các nghiên cứu về người dùng Internet, nhóm người này chiếm 26% số người dùng Internet 2020 tại Việt Nam [2], đây là nhóm người trẻ, thuộc thế hệ Z nên khả năng tiếp cận công nghệ cao hơn các nhóm khác, đây cũng là nhóm tuổi dễ bắt kịp xu hướng công nghệ, có thể đánh giá một bình luận hoặc bài viết có sắc thái như thế nào.

Tuy nhiên hạn chế của nhóm người nhỏ này là họ sẽ bị những định kiến riêng của lứa tuổi ảnh hưởng tới màu sắc của nhãn dán, các nghiên cứu tiếp theo chúng tôi sẽ đa dạng hóa hơn nhóm người gắn nhãn.

3.2. Lọc dữ liệu

Sau khi thu thập dữ liệu từ VnExpress về, chúng tôi tiến đến bước lọc dữ liệu. Đây là bước quan trọng trước khi chuyển sang tiền xử lý ngôn ngữ tự nhiên. Ở bước này, chúng tôi sẽ chuẩn hóa các từ viết tắt, những lỗi đánh máy và đặc biệt những emotion, dấu câu không cần thiết hoặc những tag html còn sót. Mục đích của công việc này nhằm làm sạch dữ liệu giúp tránh nhiễu khi trích xuất đặc trưng và huấn luyện mô hình

- Dấu câu không cần thiết, emotion và tag html: Nghiên cứu sử dụng các thư viện và những công thức Regrex để hoàn tất nhiệm vụ này, đặc biệt là chuyển tất cả văn bản thành dạng chữ thường nhằm tránh nhiễu.
- Những lỗi đánh máy: Sử dụng kỹ thuật Python nhằm chỉnh sửa những lỗi này và những lỗi sai chính tả, không đánh dấu tiếng Việt…
- Những từ viết tắt: Chúng tôi cũng thay thế những từ viết tắt và đồng nghĩa thành từ tiếng Việt tương ứng nhằm tránh lỗi và những từ OOV (Out of Vocabulary – những từ không nằm trong từ điển). Sau đây là từ điển viết tắt và đồng nghĩa của một số từ

Từ viết tắt	**Chuẩn hóa**	**Từ viết tắt**	**Chuẩn hóa**	**Từ viết tắt**	**Chuẩn hóa**
vn	việt nam	đt	điện thoại	tàu	trung quốc
mxh	mạng xã hội	đtdd	điện thoại di động	dt	điện thoại

smartphone	điện thoại thông minh	ifan	iphone	<number> củ	<number> triệu
test	kiểm tra	ip	iphone	<number> lít	<number> 00000
trâu	mạnh	ss	samsung	hịn	xịn
xài	dùng	hdh	hệ điều hành	<number> p	<number> plus
sốc	mạnh	tq	trung quốc	<number> +	<number> plus

Bảng 3-3. Bảng Từ viết tắt và đồng nghĩa

3.3. Tiền xử lý

Từ dữ liệu thô, chúng tôi sử dụng các phương pháp lọc cơ bản như điền dữ liệu còn thiếu, xóa stopword, sửa lỗi chính tả và viết tắt. Sau đó, chúng tôi trích xuất tính năng từ dữ liệu này.

3.3.1. Xử lý dữ liệu bị thiếu

Dữ liệu bị thiếu (hoặc giá trị bị thiếu) được định nghĩa là giá trị dữ liệu không được lưu trữ cho biến quan sát. Vấn đề này tương đối phổ biến trong hầu hết các nghiên cứu và có ảnh hưởng đáng kể đến kết luận rút ra từ dữ liệu [8]. Theo đó, dữ liệu bị thiếu có thể gây ra những vấn đề khác nhau. Đầu tiên, việc dữ liệu không đầy đủ làm giảm độ chính xác thống kê. Thứ hai, dữ liệu bị thiếu có thể gây ra sai lệch trong việc ước tính các tham số. Thứ ba, nó có thể làm giảm tính đại diện của các mẫu. Thứ tư, nó có thể làm phức tạp việc phân tích nghiên cứu. Mỗi vấn đề này có thể đe dọa tính chính xác của các thí nghiệm và có thể dẫn đến kết luận sai.

Hiện có 3 cách chính xử lý dữ liệu bị thiếu, bao gồm:

- Xóa dữ liệu bị thiếu: Điều này còn được biết đến như xóa hàng, loại bỏ hết tất cả các trường hợp bị thiếu
- Điền dữ liệu: Lấy trung bình của các dữ liệu xung quanh dữ liệu bị thiếu để tính ra dữ liệu đó
- Nội suy dữ liệu: Sử dụng các hàm dự đoán (thường là linear) để nội suy dữ liệu bị thiếu thông qua các dữ liệu khác

Trong nghiên cứu này, dữ liệu bị thiếu là những dữ liệu của tin tức hoặc bình luận đa số chứa hình ảnh, video. Điều này dẫn đến những dữ liệu văn bản rất ít và vô nghĩa, không có giá trị phân tích. Vì thế, trong những trường hợp này, chúng tôi chọn cách xóa dữ liệu. Chúng tôi không chọn cách nội suy dữ liệu vì đây là dữ liệu văn bản với thông tin mỗi bài khác nhau không liền mạch.

3.3.2. Xóa Stopwords

Stopwords là từ được sử dụng phổ biến và lặp lại liên tục trong văn bản (chẳng hạn như “bị”, “tôi”, “thì”, “này”, “mà”, “do”…) và đặc biệt là không mang ý nghĩa gì quá lớn hay còn gọi là những từ vô nghĩa. Để tránh mô hình bị nhiễu và dự đoán không được chính xác, chúng tôi đã loại bỏ những từ này. Chúng tôi sử dụng chỉ số TF-IDF (sẽ được nói kỹ hơn ở phần sau) để tìm ra những stopwords với threshold bằng 3.0. Sau đây là danh sách các từ ấy

bị	bởi	cả	các
cái	cần	càng	chỉ
chiếc	cho	cho	chưa
cùng	cũng	đã	đang
đây	điện_thoại	để	do

được	là	nhưng	những
rằng	rất	smartphone	tại
theo	thì	từng	vậy

Bảng 3-4. Bảng Stopwords

3.4. Trích xuất đặc trưng

Khi sử dụng thuật toán BiLSTM, LSTM và SVM, chúng tôi sử dụng 3 phương pháp trích xuất đặc trưng khác nhau, TF-IDF, Word2Vec và Bag of Words. TF-IDF phản ánh tầm quan trọng của một từ đối với kho văn bản, chúng tôi đã sử dụng thư viện keras để trích xuất đặc trưng này. Đối với Word2Vec chúng tôi đã sử dụng từ một mô hình được huấn luyện trước [29], mỗi từ là số thứ i của từ đó trong từ điển Word2Vec. Hơn nữa, Bag of Words cũng được sử dụng như là một trong những phương pháp xử lý ngôn ngữ tự nhiên cơ bản nhất, các câu là một dãy số đại diện các từ trong câu đó. Sau đây là chi tiết về những phương pháp này

3.4.1. Bag of Words

Bag of Words còn được gọi là "túi từ", một phương pháp trích xuất đặc trưng cơ bản trong xử lý văn bản. Ý tưởng của phương pháp này sẽ tách toàn bộ những từ hoặc cụm từ trong 1 văn bản (tokenize). Trong nghiên cứu này chúng tôi tách từ trong văn bản ở dạng một từ (unigram) hoặc 1 cụm tối đa là 2 từ (bigram).

Với mỗi văn bản, ta sẽ tạo ra một vector đặc trưng có số chiều bằng số từ/cụm từ đã tách, mỗi phần tử đại diện cho số từ tương ứng xuất hiện trong văn bản đó. Ví dụ với hai câu như sau:

- "Điện thoại dễ hết pin"
- "Pin yếu. Không có cái nào pin trâu cả"

Với hai câu trên, chúng tôi sẽ tách thành túi từ gồm: “điện_thoại”, “dễ”, “hết”, “pin”,”yếu”, “không_có”, “cái”, “nào”, “trâu”, “cả”

Thì vào lúc này hai vector đặc trưng tương ứng của chúng là:

- [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
- [0, 0, 0, 2, 1, 1, 1, 1, 1, 1]

Với phương pháp này vẫn còn tồn tại một số nhược điểm như số lượng vector quá dài và thừa thải (những giá trị 0 quá nhiều) dẫn đến dung lượng cần sử dụng quá nhiều với độ dài vector có thể lên hàng triệu, chục triệu cho một câu và đặc biệt không hiển thị được thứ tự của các từ cũng như sự quan trọng của chúng đối với ngữ cảnh văn bản. Ví dụ với hai câu “Pin này không yếu” và “Pin này yếu không” sẽ trả lại cùng một kết quả nếu dùng phương pháp này.

3.4.2. TF-IDF

TF-IDF (Term frequency – Inverse document frequency) được xem là trọng số của một từ trong văn bản thể hiện mức độ quan trọng của từ này trong câu, đoạn văn cũng như trong toàn bộ văn bản.

TF (Term frequency): Tần suất xuất hiện của một từ trong văn bản. Nhưng trong thực tế có những văn bản độ dài khác nhau và những văn bản dài thường sẽ có nhiều từ xuất hiện hơn. Vì thế, tần suất này được chia cho độ dài của văn bản nhằm chuẩn hóa (normalization).

TF được tính bởi công thức: $tf(t) = \frac{f(t,d)}{T}$

(với t là một từ trong đoạn văn bản; $f(t,d)$ là tần suất xuất hiện của t trong đoạn văn bản d; TT là tổng số từ trong đoạn văn bản d).

IDF (Inverse document frequency): Đây là số liệu đo độ quan trọng của một từ trong văn bản. Khi sử dụng TF, mỗi từ đều quan trọng như nhau, nhưng

có một số từ mà chúng ta gọi là Stopwords xuất hiện nhiều nhưng lại đóng vai trò ít quan trọng. Vì vậy, chúng ta cần một phương thức loại bỏ những từ này và tăng độ quan trọng của những từ ít xuất hiện nhưng có ý nghĩa đặc biệt cho đoạn văn bản

IDF được tính bởi công thức: $idf(t) = log \frac{N}{|t \in D : t \in d|}$

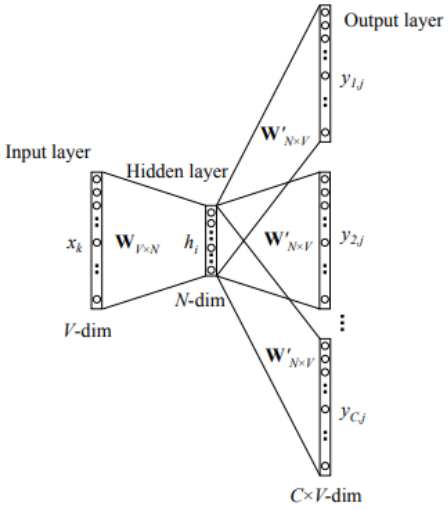
(trong đó N là tổng số văn bản và $|t \in D: t \in d|$ là số văn bản chứa từ N)

Chỉ số TF-IDF là tích của hai thông số này: $tf\_idf(t) = tf(t) \times idf(t)$

Thông số trong nghiên cứu này ngoài trích đặc trưng cũng được sử dụng để xác định stopwords với threshold bằng 3.0 theo đó những từ xuất hiện nhiều lần trong nhiều văn bản lặp đi lặp lại sẽ bị loại bỏ.

3.4.3. Word2Vec

Thay vì phải mã hóa từng từ và cụm từ theo một con số nhất định, điều này không biểu thị được mối quan hệ giữa chúng với nhau. Vì thế, đã xuất hiện một cách mã hóa từ thành vecto mới có tên là Word Embedding, ở đó các từ sẽ có mối quan hệ với nhau về ngữ nghĩa ví dụ như nam-nữ, anh-chị với kích thước NxD với N là kích thước văn bản và D là số chiều vector.



Hình 3-3. Mô hình Skip-gram trong word2vec [28]

Theo Xin Rong [28], đầu vào của mô hình sẽ là $x_1, x_2, \dots, x_k$ và weight matrix (trọng số) giữa đầu vào và hidden layer sẽ là ma trận W_{VxN} với V là số lượng văn bản và N là số chiều đại diện cho 1 từ gọi là v_w (hay còn gọi là vector của 1 từ có N chiều). Sau đó, chúng ta tính hidden layer sẽ bằng:

$$h = W^T \times x = v_w^T$$

Tương tự như vậy, hidden layer đến đầu ra cũng sẽ là $W' = w'_{i,j}$ với i,j là hàng và cột tương ứng trong W'. Từ đó, tương tự như trên ta dễ dàng tính được trọng số đầu ra.

$$u_j = v'_{w_j} \times h$$

Cuối cùng, chúng ta dùng hàm softmax để chuyển thành số giữa 0 và 1 cho từ đó.

$$P(w_j|w_I) = y_i = \frac{e^{u_j}}{\sum_{k=1}^{V} e^{u_k}}$$

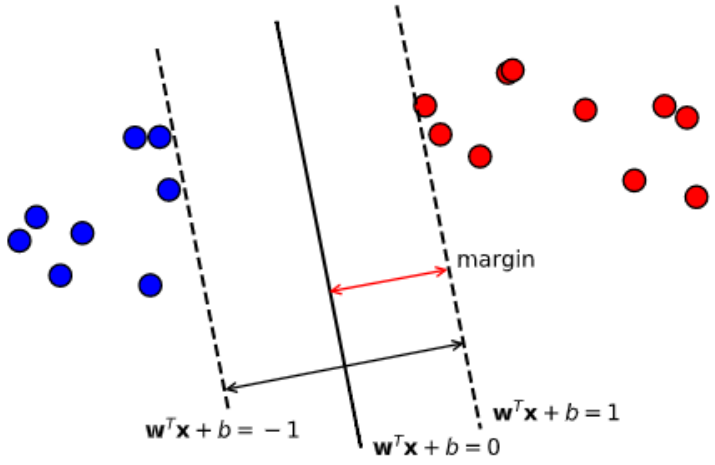
Trong nghiên cứu này, chúng tôi sử dụng pre-trained model word2vec tiếng Việt [29] với chuyển 1 từ thành 1 vector có 300 chiều để trích xuất đặc trưng.

3.5. Training model

3.5.1. SVM

Support Vector Machine (SVM) là một mô hình phân loại hoạt động bằng việc xây dựng một siêu phẳng (hyperplane) có (n - 1) chiều trong không gian n chiều (với n là số lượng đặc trưng) của dữ liệu sao cho siêu phẳng này phân loại các lớp một cách tối ưu nhất. Nói cách khác, cho một tập dữ liệu có nhãn (học có giám sát), thuật toán sẽ dựa trên dữ liệu học để xây dựng một siêu phẳng tối ưu được sử dụng để phân loại dữ liệu mới. Ở không gian 2 chiều thì siêu phẳng này là 1 đường thẳng phân cách chia mặt phẳng không gian thành 2 phần tương ứng 2 lớp với mỗi lớp nằm ở 1 phía của đường thẳng.

Bài toán của SVM là phải tối đa hóa khoảng cách giữa siêu phẳng và các điểm dữ liệu (margin)



Hình 3-4. Max margin của SVM (Nguồn: dominhhai.github.io)

Để xác định được siêu phẳng này, đầu tiên phải xác định các điểm gần với siêu phẳng mà trong trường hợp này là 1:

$$min|W^T x + b|=1$$

Tiếp theo, ta cần tính khoảng cách giữa các điểm dữ liệu sao cho tìm được khoảng cách xa nhất để xác định vị trí siêu phẳng. Ta tính khoảng cách đó như sau:

$$p = min\frac{|w^T x + b|}{||w||} = \frac{1}{||w||}$$

Do đó, để p đạt giá trị lớn nhất thì đồng nghĩa $||w||$ đạt giá trị nhỏ nhất và các điểm $y_i(w^T x + b) \geq 1$:

$$(w, b) = \arg\min_{w,b}\frac{1}{2}||w||^2 \; với \; y_i(w^T x + b) \geq 1, i \in [1, m]$$

Trong đó m là các điểm dữ liệu và lấy bình phương và chia đôi để tiện trong việc tối ưu lồi và đạo hàm. Sau đó, công thức được đưa về bài toán đối ngẫu và sử dụng phương pháp Lagrance. Ta có λ được tính như sau

$$\lambda = \arg\max_{\lambda}\sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i,j=1}^{m}\lambda_i\lambda_j y_i y_j x_i^T x_j$$

Với $\lambda_i \geq 0 \; \wedge \sum_{i=1}^{m}\lambda_i y_i = 0, i \in [1, m]$

Tiếp theo đó, chúng ta giải phương trình tìm λ và nhận được:

$$w = \sum_{i=1}^{m}\lambda_i y_i x_i$$

$$b = y_i - \sum_{j=1}^{m}\lambda_i y_i x_i x_j^T$$

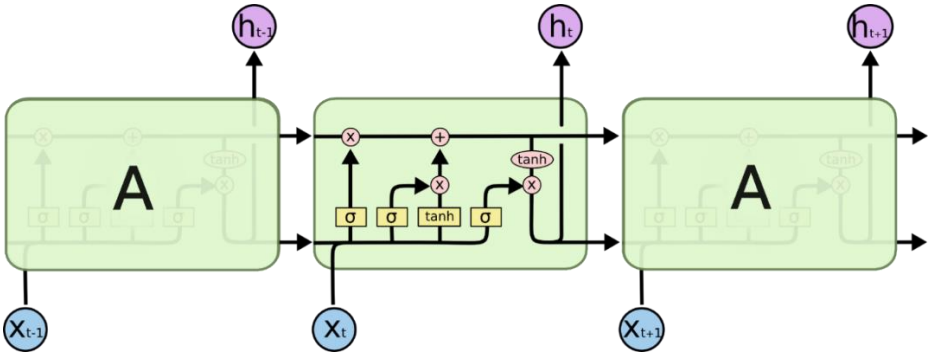
Trong đó, (x_i, y_i) là điểm dữ liệu nằm trên đường gần nhất với siêu phẳng (biên siêu phẳng). Theo đó, gọi S là tập hợp của tất cả những điểm này, ta tính b bằng tổng tất cả b_i. Sau đó biểu thức của ta sẽ là:

$$w^T + b = \sum_{i=1}^{m} \lambda_i y_i x_i^T x + \frac{1}{|S|} \sum_{i \in S} (y_i - \sum_{j=1}^{m} \lambda_i y_i x_i x_j^T)$$

3.5.2. LSTM

Mỗi cấu trúc mạng hồi quy (Recurrent Neural Network) đều là sự lặp lại module của các mạng neuron với đầu ra của module này sẽ là đầu vào của module khác. Thường các mạng RNN chỉ có một biến đầu vào và một đầu ra, chính vì lý do đó nên các mạng RNN sẽ không giải quyết được vấn đề phụ thuộc xa, điều đó mang ý nghĩa mô hình sẽ "quên" dần và không xử lý được những văn bản dài vì mô hình không nhớ được toàn bộ các đặc trưng từ đầu đến cuối

Và mô hình LSTM ra đời để giải quyết điều này. Với thiết kế phức tạp hơn bao gồm trạng thái tế bào, các cổng quên (forget gate).



Hình 3-5. Kiến trúc LSTM (Nguồn: dominhhai.github.io)

Đầu tiên, với đầu vào là x_t và h_{t-1} từ module trước đó, LSTM sẽ quyết định thông tin nào sẽ giữ lại qua forget gate trả giá trị từ [0,1] (0: bỏ đi toàn bộ, 1: giữ lại toàn bộ) với hàm sigmoid được thể hiện theo công thức sau:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f)$$

Tiếp theo, LSTM xem xét thông tin nào sẽ thêm vào dựa vào input gate. Quá trình này gồm 2 phần, phần đầu cũng sử dụng hàm sigmoid để trả về từ [0,1] và phần thứ hai chuyển qua hàm tanh nhằm tạo ra vector mới C_t'. Sau đó chúng ta kết hợp 2 phần này nhằm tạo ra giá trị thêm vào

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i$$

$$C_t' = \tanh\,(W_C[h_{t-1}, x_t] + b_C)$$

Cuối cùng, chúng ta kết hợp lọc thông tin bị "quên" và thông tin thêm mới vào để cập nhật trạng thái tế bào C_t

$$C_t = f_t \times C_{t-1} + i_t \times C_t'$$

Tiếp theo là tính đầu ra, ta chuyển trạng thái tế bào mới này qua một hàm tanh để biến đổi giá trị trong khoảng [-1,1] và nhân với sigmoid (hàm sigmoid nhằm xác định phần nào của module này muốn xuất ra)

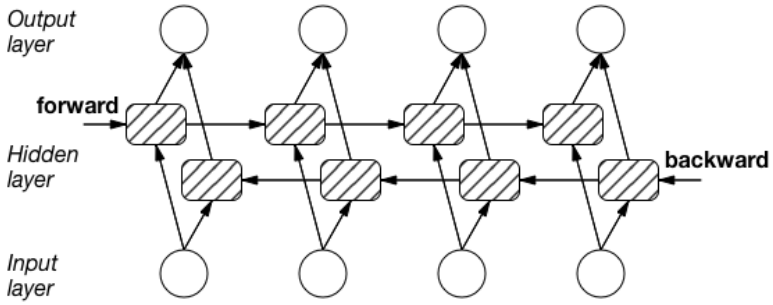
$$\sigma_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \times \tanh\,(C_t)$$

3.5.3. BiLSTM

Về cơ bản, BiLSTM là nâng cấp của mô hình LSTM với cấu trúc tương tự. Sự khác biệt lớn nhất giữa chúng là BiLSTM cho phép duyệt văn bản theo cả hai chiều. Để hiểu sâu hơn về một đoạn văn hay một câu văn thì không chỉ vào các thông tin phía trước của từ đang xét mà còn cả các thông tin phía sau nhưng LSTM chỉ là mô hình forward (nghĩa là từ trái sang phải) và chỉ có thể dự đoán nhãn của từ hiện tại dựa trên thông tin có được từ các từ nằm trước đó. Và Bidirectional LSTM (BiLSTM) đã được tạo ra để khắc phục điểm yếu trên. Một mô hình BiLSTM thường chứa 2 mạng LSTM đơn được sử dụng đồng thời và độc lập để

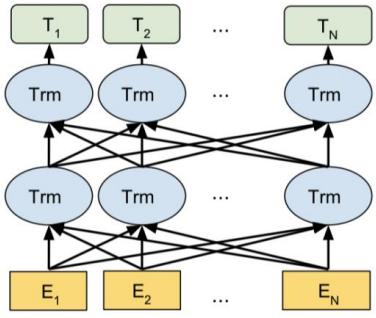
mô hình hoá chuỗi đầu vào theo 2 hướng: từ trái sang phải (forward) và từ phải sang trái (backward). Mô hình BiLSTM được thể hiện rõ theo hình dưới đây:



Hình 3-6. Kiến trúc BiLSTM (Nguồn: medium.com)

3.5.4. BERT

BERT là một mô hình ngôn ngữ được tạo ra bởi các nhà nghiên cứu tại Google dựa trên transformers. BERT là mô hình được huấn luyện trước với dữ liệu văn bản lớn (Wikipedia, v.v.) với mục đích dễ dàng thực hiện tinh chỉnh để điều chỉnh mô hình những bài toán và vấn đề vấn đề cụ thể. Sơ đồ kiến trúc BERT được hiển thị trong hình sau:

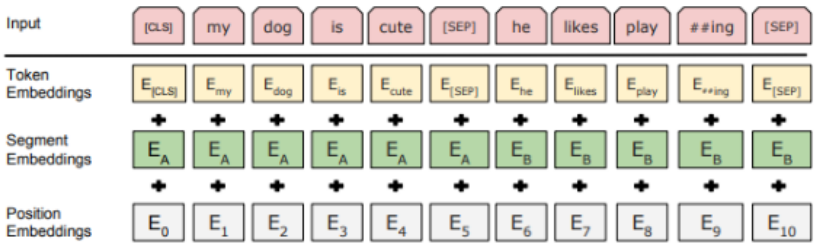


Hình 3-7. Kiến trúc của BERT [12]

Ngoài ra, BERT chính là một Transformer 2 chiều thường được gọi là Transformer encoder trong khi các phiên bản Transformer chỉ sử dụng ngữ cảnh bên trái thường được gọi là Transformer decoder vì nó có thể được sử

dụng để tạo ra văn bản, và vì thế, BERT có thể xem xét nhiều ngữ cảnh trong một đoạn văn hơn.

Đầu vào của BERT có nhiều điểm khác biệt so với những mô hình còn lại, cụ thể nguyên lý đầu vào của BERT như sau:



Hình 3-8. Đầu vào BERT [12]

Khi có một chuỗi đầu vào cụ thể, biểu diễn đầu vào của chúng được xây dựng bằng cách tính tổng các token với vector phân đoạn và vị trí tương ứng của các từ trong chuỗi. Đặc biệt, BERT sử dụng WordPiece với từ điển cụ thể tùy theo loại BERT để phân tách các từ và dùng dấu ## làm ký hiệu cho điều này.

Có 2 cách chính để huấn luyện mô hình cho BERT: Mask LM và Next sentence prediction, vì đây là bài toán phân lớp nên trong nghiên cứu này, chúng tôi chỉ tập trung vào Mask LM. Ta có thể nhận thấy dễ dàng mô hình Deep Learning dựa trên ngữ cảnh 2 chiều (từ trái sang phải và từ phải sang trái) mạnh mẽ so với mô hình ngữ cảnh 1 chiều

Tuy vậy, lý do từ trước không thể dùng mô hình 2 chiều do một từ có thể gián tiếp "nhận biết" chính nó thông qua nhiều lớp, điều này gây nhiễu trong huấn luyện mô hình.

Để huấn luyện một mô hình tìm ra vector của một từ nào đó dựa vào ngữ cảnh 2 chiều, chúng ta bắt buộc phải che giấu đi một số token đầu vào ngẫu nhiên và sau đó chỉ dự đoán các token được giấu đi đó và điều này gọi là

Masked LM (MLM). Trong trường hợp này, các hidden vectors ở lớp cuối cùng tương ứng với các tokens được ẩn đi sẽ được đưa vào 1 lớp softmax để dự đoán.

Tuy nhiên, điều này lại gây ra 2 vấn đề khác nhau. Đầu tiên sự không phù hợp giữa việc pre-train (huấn luyện trước) và fine-tuning (tinh chỉnh) vì các token được [MASK] không bao giờ được nhìn thấy trong quá trình tinh chỉnh. Để giảm thiểu điều này, không phải lúc nào chúng ta cũng thay thế các từ được giấu đi bằng token [MASK]. Thay vào đó, BERT sẽ chọn 15% tokens một cách ngẫu nhiên và thực hiện các bước như sau:

Ví dụ với câu: "điện_thoại xịn quá" Từ được chọn để mask là từ "xịn".

Thay thế 80% từ được chọn trong dữ liệu huấn luyện thành token [MASK] → "điện_thoại [MASK] quá"

10% sẽ được thay thế bởi 1 từ ngẫu nhiên. => "điện_thoại điên quá"

10% còn lại được giữ không thay đổi –> "điện_thoại xịn quá "

Và vì thế mô hình không biết được từ nào đạt yêu cầu dự đoán hoặc từ nào đã được thay thế bằng một từ ngẫu nhiên, do đó, bắt buộc nó phải giữ một vector theo ngữ cảnh của mỗi token đầu vào. Ngoài ra, do thay thế số lượng ít ($1.5\% = 15\% \times 10\%$) các tokens bằng một từ ngẫu nhiên nên điều này dường như sẽ không làm ảnh hưởng tới khả năng hiểu ngôn ngữ của mô hình.

Nhược điểm thứ 2 của việc sử dụng MLM là chỉ có 15% tokens được dự đoán trong mỗi lần (mỗi batch), điều này mang ý nghĩa là cần thêm các bước sử dụng các pre-trained model (mô hình huấn luyện trước) khác để mô hình hội tụ và đạt được độ chính xác cao hơn.

3.6. Kết hợp đầu ra mô hình

Cuối cùng, chúng tôi sử dụng kết quả đầu ra của hai mô hình để phân loại tin tức. Trước hết, chúng tôi nhân đầu ra mô hình dự đoán tin tức với trọng số được tính bằng cách chia tổng số bình luận tích cực và tiêu cực cho số lượng bình luận trung lập. Điều này mang ý nghĩa nếu nội dung tin tức càng thu hút nhiều người thảo luận về sản phẩm thì càng quan trọng hơn. Ngoài ra, chúng tôi cộng 1 vào mẫu số của trọng số vì một số tin tức không có bình luận trung lập và trọng số sẽ không tính được. Cuối cùng, chúng tôi đã cộng tổng số đầu ra của mô hình dự đoán bình luận với N là tổng số bình luận cho nội dung bài báo/tin tức đó. Nếu $result < 0$, bản tin sẽ tiêu cực, nếu $result > 0$, bản tin sẽ tích cực và nếu $result = 0$, bản tin sẽ không gây tác động nào.

$$result = \frac{num_{pos} + num_{neg}}{num_{neutral} + 1} \times output_{news} + \sum_{i=1}^{N} output_{comment}$$

CHƯƠNG 4. KẾT QUẢ THỬ NGHIỆM

4.1. Dữ liệu

Tóm lược nội dung ở trên, nghiên cứu này sử dụng bộ dữ liệu bao gồm hơn 2500 bài báo tin tức và hơn 8700 bình luận được lấy từ VnExpress trong lĩnh vực điện thoại thông minh được dán nhãn là 1 (tích cực), 0 (trung tính) hoặc 2 (tiêu cực). Tin tức cũng bao gồm hình ảnh, video nhưng trong nghiên cứu này, chỉ có nội dung tin tức và bình luận dưới dạng văn bản được sử dụng.

4.2. Mô hình

Chúng tôi sử dụng Python trên nền tảng Google Colab và Tensorflow phiên bản 1 và 2. Ngoài ra, chúng tôi cũng sử dụng thư viện sklearn, keras và bert để hoàn tất xây dựng các mô hình. Sau đây là chi tiết mô hình.

4.2.1. SVM

Chúng tôi sử dụng phân loại Support Vector Machine do sklearn cung cấp để huấn luyện và đánh giá mô hình. Loại kernel được sử dụng trong thuật toán này là “linear” và “rbf” (C và gamma) với các tham số khác nhau.

Ngoài ra, trong mô hình SVM, nghiên cứu cũng sử dụng dữ liệu trích xuất vector từ Bag of Words, TF-IDF và Word2Vec

4.2.2. LSTM

Với mô hình LSTM, nghiên cứu sử dụng thư viện keras của Tensorflow và mô hình tối ưu Adam với learning rate (tốc độ học) là tham số thí nghiệm. Chúng tôi xây dựng mô hình với 4 lớp: 1 lớp LSTM với 128 units và 3 lớp Dense tương

ứng với 64 và 32 units. Lớp Dense cuối cùng chính là lớp softmax xuất ra ba xác suất tương ứng với 1 (tích cực), 0 (trung tính) và 2 (tiêu cực)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         (None, 300)               0
_________________________________________________________________
reshape_5 (Reshape)          (None, 10, 30)            0
_________________________________________________________________
lstm_5 (LSTM)                (None, 128)               81408
_________________________________________________________________
dense_13 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_14 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_15 (Dense)             (None, 3)                 99
=================================================================
Total params: 91,843
Trainable params: 91,843
Non-trainable params: 0
```

Hình 4-1. Mô hình LSTM

4.2.3. BiLSTM

Với mô hình BiLSTM, chúng tôi cũng xây dựng tương tự với mô hình LSTM là sử dụng thư viện keras của Tensorflow và mô hình tối ưu Adam với learning rate (tốc độ học) là tham số thí nghiệm. Mô hình này cũng giống với mọ hình LSTM gồm 4 lớp: 1 lớp BiLSTM với 256 units và 2 lớp Dense tiếp theo tương ứng với 64 và 32 units. Lớp Dense cuối cùng chính là lớp softmax xuất ra ba xác suất tương ứng với 1 (tích cực), 0 (trung tính) và 2 (tiêu cực).

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 300)               0
_________________________________________________________________
reshape_2 (Reshape)          (None, 10, 30)            0
_________________________________________________________________
bidirectional_2 (Bidirection (None, 256)               162816
_________________________________________________________________
dense_4 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_5 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 99
=================================================================
Total params: 181,443
Trainable params: 181,443
Non-trainable params: 0
```

Hình 4-2. Mô hình BiLSTM

4.2.4. BERT

Trong mô hình BERT, khác với 3 mô hình còn lại, chúng tôi không sử dụng đặc trưng được trích xuất từ ba phương pháp mà sử dụng thẳng dữ liệu văn bản cho đầu vào. Ngoài ra, với mô hình pre-trained, nghiên cứu sử dụng BERT Mutilingual Cased hỗ trợ 104 ngôn ngữ trong đó có cả tiếng Việt với 12-layer, 768-hidden, 12-heads, 110M parameters.

4.3. Thang đo

4.3.1. Accuracy (Độ chính xác):

Đây là thang đo đơn giản nhất được đo bằng tỉ lệ dự đoán đúng và tổng số dự đoán trong dữ liệu test

4.3.2. True Positive, True Negative, False Positive, False Negative

Trong bài toán phân loại, sẽ có những thông số đóng vai trò quan trọng hơn những thông số còn lại và chúng ta cần sử dụng nhiều thang đo khác nhau để đo lường điều này. Cụ thể, trong nghiên cứu này, thà nhầm lẫn tích cực thành tiêu cực còn hơn bỏ sót những bình luận tiêu cực. Vì thế, chúng tôi chia thành các thông số như sau (chúng tôi xem trọng yếu tố dự đoán tiêu cực vì sẽ nhận được nhiều lời góp ý của khách hàng cho sản phẩm hơn):

- True Positive (TP): dự đoán tiêu cực đúng
- True Negative (TN): dự đoán tích cực đúng
- False Positive (FP): nhầm lẫn tích cực thành tiêu cực
- False Negative (FN): bỏ sót tiêu cực

	Dự đoán		
Trong thực tế		**Negative**	**Positive**
	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Bảng 4-1. Bảng mô tả thang đo

4.3.3. Precision:

Precision là một thang đo nhằm xác định khi mà dự đoán sai các Positive (bình luận tiêu cực) rất nguy hiểm và gây ảnh hưởng lớn. Công thức của Precision được tính như sau:

$$precision = \frac{TP}{TP + FP}$$

4.3.4. Recall:

Recall thể hiện rằng bao nhiêu mẫu positive trong thực tế được xác định đúng. Thông số này dùng để đánh giá 1 mô hình khi mà việc dự đoán sai 1 mẫu positive trong thực tế rất nguy hiểm.

$$recall = \frac{TP}{TP + TN}$$

F1-Score: Là thông số cân bằng giữa Precision và Recall. F1 Score khác với Accuracy ở điểm nếu bài toán có quá nhiều mẫu Negative sẽ dẫn đến sự sai lệch trong thang đo này. Vì dữ liệu này cũng có nhiều mẫu Negative (bình luận tích cực) nên trong nghiên cứu này, chúng tôi chọn F1 score làm thang đo chính.

4.4. Kết quả và thảo luận

Mô hình dự đoán tin tức hiệu quả nhất là BERT với F1-Score 74% và mô hình dự đoán nhận xét bình luận hiệu quả nhất là BiLSTM với F1-Score 71%. Ngoài ra, TF-IDF là phương pháp hiệu quả nhất để trích xuất vectơ trong cả hai mô hình. Điều này hợp lý vì nội dung tin tức đã được kiểm duyệt cẩn thận, không có lỗi đánh máy và ngữ pháp cũng như có ngữ cảnh rõ ràng, và vì thế BERT là mô hình tốt nhất cho việc này vì tính năng tự truy xuất đặc trưng hiệu quả cho văn bản dài. Bên cạnh đó, nhận xét bình luận có độ dài ngắn và nội dung đa dạng hơn nên BiLSTM là lựa chọn tốt nhất. Kết quả chi tiết được thể hiện trong Bảng 1.

News Prediction Model							
Learning Rate		1e-5	5e-5	1e-4	5e-4	1e-3	5e-3
LSTM	TF-IDF:	0.47	0.59	0.60	0.63	0.64	0.64
	Word2vec:	0.36	0.49	0.51	0.53	0.59	0.61
	BoW:	0.41	0.54	0.56	0.59	0.60	0.63
BiLSTM	TF-IDF:	0.61	0.63	0.65	0.65	0.70	0.72
	Word2vec:	0.43	0.54	0.62	0.62	0.65	0.66
	BoW:	0.45	0.56	0.58	0.63	0.68	0.70
BERT		0.62	0.65	0.68	0.71	0.71	0.74

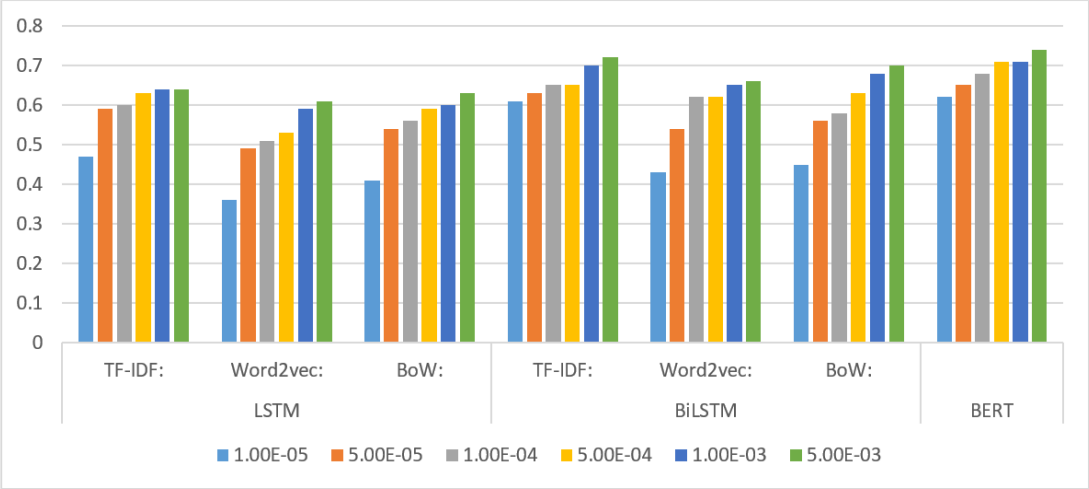
SVM News Prediction Model									
	Radial Basic Function (RBF)								Linear
Extract Vectors	C	1e-3	5e-3	1e-2	5e-2	1e-1	5e-1	1e	None

TF-IDF	10	0.25	0.27	0.40	0.51	0.55	0.54	0.52	0.51
	100	0.41	0.52	0.54	0.54	0.56	0.52	0.51	0.52
	1000	0.54	0.51	0.52	0.54	0.55	0.51	0.51	0.52
Word2Vec	10	0.34	0.36	0.39	0.43	0.48	0.53	0.52	0.49
	100	0.37	0.41	0.45	0.50	0.51	0.52	0.49	0.51
	1000	0.45	0.51	0.51	0.53	0.52	0.52	0.51	0.51
Bag of Words	10	0.51	0.56	0.52	0.43	0.40	0.24	0.23	0.54
	100	0.57	0.56	0.53	0.45	0.41	0.23	0.23	0.54
	1000	0.59	0.53	0.54	0.42	0.41	0.24	0.22	0.55

Comment Prediction Model							
Lear		1e-5	5e-5	1e-4	5e-4	1e-3	5e-3
LSTM	TF-IDF:	0.54	0.55	0.58	0.58	0.59	0.60
	Word2vec:	0.46	0.50	0.53	0.53	0.55	0.58
	BoW:	0.32	0.41	0.44	0.49	0.52	0.54
BiLSTM	TF-IDF:	0.56	0.56	0.60	0.65	0.68	0.71
	Word2vec:	0.42	0.50	0.52	0.57	0.62	0.66
	BoW:	0.35	0.42	0.47	0.51	0.57	0.59
BERT		0.60	0.61	0.64	0.64	0.67	0.70

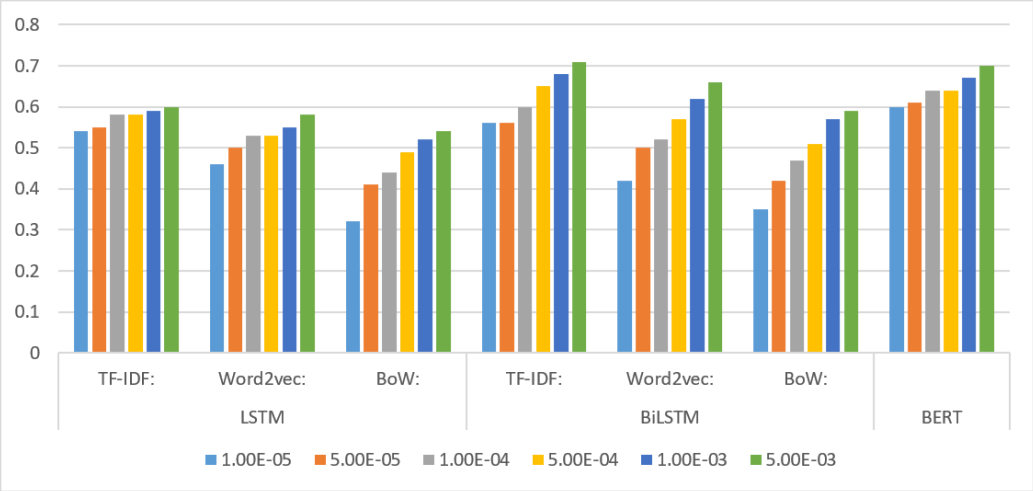
SVM Comment Prediction Model									
	Radial Basic Function (RBF)								Linear
Extract Vectors	C	1e-3	5e-3	1e-2	5e-2	1e-1	5e-1	1e	None
TF-IDF	10	0.26	0.30	0.39	0.51	0.54	0.53	0.50	0.52
	100	0.38	0.52	0.54	0.53	0.54	0.55	0.51	0.53
	1000	0.55	0.54	0.52	0.53	0.53	0.54	0.50	0.52
Word2Vec	10	0.32	0.35	0.42	0.45	0.49	0.51	0.48	0.48
	100	0.34	0.39	0.47	0.49	0.50	0.53	0.51	0.51
	1000	0.39	0.50	0.50	0.51	0.50	0.52	0.49	0.51
Bag of Words	10	0.43	0.52	0.53	0.47	0.39	0.25	0.24	0.51
	100	0.52	0.53	0.54	0.47	0.40	0.25	0.23	0.50
	1000	0.53	0.51	0.52	0.48	0.41	0.26	0.24	0.51

Bảng 4-2. Bảng kết quả chi tiết



Hình 4-3. Biểu đồ phân loại tin tức

Ngoài ra, đối với biểu đồ phân loại tin tức như trên, ta thấy rõ độ chính xác của BiLSTM và BERT không khác nhau mấy với mọi learning rate. Tuy vậy, khi ở mô hình BiLSTM, khi learning rate bằng 10^{-4}, ta thấy độ chính xác của mô hình sử dụng Word2Vec cao hơn mô hình Bag of Words nhưng chung quy thì mô hình sử dụng Word2Vec vẫn có độ chính xác thấp hơn mô hình Bag of Words. Điều này có thể lý giải do mô hình Word2Vec sử dụng số lượng vector quá lớn dẫn đến đặc trưng quá nhiều và lượng dữ liệu lại không nhiều dẫn đến tỷ lệ thấp hơn so với Bag of Words.



Hình 4-4. Biểu đồ phân loại bình luận

Đối với biểu đồ phân loại bình luận người dùng, chúng ta thấy sự khác biệt rõ ràng khi mô hình sử dụng Word2Vec lại có độ chính xác nhìn chung cao hơn mô hình sử dụng Bag of Words. Sở dĩ xảy ra sự khác biệt này là do bình luận người dùng ngắn hơn, sử dụng những từ ngữ lặp lại nhiều và ít phong phú hơn. Điều này góp phần làm Word2Vec có ít đặc trưng và nhiều dữ liệu để dự đoán chính xác hơn. Ngoài ra, với learning rate nhỏ hơn 5.10-5 thì độ chính xác của mô hình BERT cao hơn hẳn những mô hình còn lại nhưng khi learning rate tăng lên thì độ chính xác tăng không đáng kể, điều này cũng do độ dài của bình luận quá ngắn ảnh hưởng đến đặc trưng mà BERT có thể trích xuất, ngược lại BiLSTM với khả năng duyệt hai chiều đã nhanh chóng tăng độ chính xác và là mô hình tốt nhất cho dự đoán bình luận.

CHƯƠNG 5. KẾT LUẬN

5.1. Ưu điểm

Để giải quyết bài toán "Khai phá ý kiến người dùng thông qua tin tức trực tuyến trong lĩnh vực điện thoại thông minh bằng Machine Learning", nhóm nghiên cứu đã thu thập dữ liệu bình luận của người dùng, nội dung bài viết trên trang VnExpress sau đó áp dụng một số mô hình để tìm ra các mô hình phù hợp nhất cho bài toán này:

Trong nghiên cứu này, chúng tôi sử dụng hai mô hình để dự đoán toàn bộ một bài báo. Mỗi mô hình áp dụng bốn phương pháp (SVM, LSTM, BiLSTM, BERT) nhằm tìm ra phương pháp nào hoạt động hiệu quả, độ chính xác nhất tương ứng với từng mô hình và đạt được kết quả tối ưu từ tập dữ liệu này. Không những thế, ở thời đại ngày nay càng có nhiều người chia sẻ suy nghĩ và cảm xúc của bản thân cũng như kinh nghiệm trên các diễn đàn trực tuyến, đây là một cơ hội tốt nghiên cứu tiếp tục phát triển sâu và ứng dụng rộng rãi hơn.

Bằng việc xử lý Python và thu thập dữ liệu liên tục từ Internet trong nhiều tuần, nhóm đã thu thập được dữ liệu về bài báo của hơn 5 sản phẩm thuộc nhiều dòng điện thoại khác nhau, từ trung cấp, cận trung cấp đến cao cấp thuộc nhiều thương hiệu (Samsung, Lenovo, Oppo …) với hơn 8.700 bình luận và 2500 bài báo từ trang VnExpress.net liên quan đến chủ đề này. Sau khi tiền xử lý dữ liệu, nhóm đã tiến hành chuẩn hóa dữ liệu và gán nhãn cho mỗi bình luận/bài viết theo trạng thái tích cực, tiêu cực hoặc trung tính. Công đoạn này hoàn toàn thực hiện thủ công tiêu tốn nhiều thời gian, đòi hỏi sự nhẫn nại và tỉ mỉ, cuối cùng nhóm đưa những dữ liệu này vào các mô hình và cho ra những kết quả khả quan.

5.2. Hạn chế

Tuy đạt được những kết quả tích cực, nhóm nhận thấy nghiên cứu này còn nhiều mặt hạn chế như thiếu tính khách quan trong giai đoạn gán nhãn dữ liệu. Việc gán nhãn cảm xúc tích cực hay tiêu cực cho các dòng dữ liệu đều được thực hiện thủ công, do đó bước này có thể bị ảnh hưởng suy nghĩ chủ quan của các thành viên, dẫn tới tình trạng thiếu tính khách quan vì phụ thuộc vào cảm xúc, góc nhìn của một người cụ thể đang thực hiện gắn nhãn thời điểm đó.

Chúng tôi vẫn gặp nhiều khó khăn trong việc chuẩn hóa tiếng Việt, xây dựng từ điển cho các từ viết tắt, chỉnh sửa lỗi chính tả, lọc ý nghĩa, từ ghép và stopword cho lĩnh vực điện thoại. Chưa thể áp dụng giải thuật một cách tự động để hạn chế tối đa những từ vô nghĩa, ít phổ biến gây ảnh hưởng đến quá trình "máy học" mà kết quả bị sai lệch.

Số lượng các nhãn còn hạn chế vì chỉ tập trung phân loại 3 nhãn là "tích cực", "tiêu cực" và "trung tính", trong khi thực tế cảm xúc của người dùng còn đa dạng hơn nữa như: vui, buồn, giận dữ, ngạc nhiên, ghét, sợ hãi, thích,…

Dữ liệu chỉ mới tập trung vào một trang VnExpress.net, tuy là một trang tin tức rất lớn ở Việt Nam, tuy nhiên nguồn tin chỉ từ một nguồn dẫn đến kết quả có thể mang các yếu tố phi kĩ thuật như chínht rị, định hướng dư luận, mục đích của trang báo trong giai đoạn đó, ...

5.3. Hướng phát triển trong tương lai

Một số hướng phát triển của bài báo này có thể áp dụng vào tương lai như:

Ứng dụng kết quả của bài báo trong việc đa dạng hóa nguồn tin, đa dạng hóa ngôn ngữ, từ đó có thể giúp được cho các doanh nghiệp, các công ty, phòng ban Marketing có thể có một công cụ nhằm dự đoán, phân tích được thị hiếu của thị

trường, xử lý khủng hoảng truyền thông, gom ý kiến cho phòng ban R&D nâng cấp và cải tiến sản phẩm sau này.

Nhận thấy kết quả của bài viết ứng dụng tốt trong việc phân tích dữ liệu mạng xã hội, các trang tin tức, từ đó có thể tạo nên các mô hình dự đoán tin giả (fake news) - một thứ mà trong thời kì COVID-19 cũng đang bùng lên như một đại dịch thứ hai.

Kết quả của nghiên cứu một lần nữa khẳng định lại, dữ liệu của các trang tin tức đóng vai trò rất lớn trong việc định hướng dư luận, định hướng sự ảnh hưởng của nhãn hàng, đây có thể là một hướng đi rất tốt cho các doanh nghiệp khi họ muốn khảo sát thị trường, phát triển sản phẩm, không chỉ dựa vào định tính đơn thuần mà dựa vào định lượng từ tập dữ liệu khổng lồ trên Internet.

Bên cạnh đó, trong tương lai xa hơn, chúng tôi có thể ứng dụng kết quả bài nghiên cứu để xây dựng nên một hệ thống đánh giá và phân loại tin tức thời gian thực, cơ chế hoạt động là liên tục thu thập dữ liệu, áp dụng mô hình và đưa ra các báo cáo trực quan, hỗ trợ trực tiếp cho doanh nghiệp. Cụ thể với nhà sản xuất, công cụ có thể giúp họ nhận thấy được các tin tức nào đang bình luận về chất lượng sản phẩm của họ, khách hàng đang nói gì về sản phẩm, họ đã làm tốt và chưa tốt điều gì về sản phẩm, từ đó giúp cho việc cải tiến sản phẩm được kịp thời và tiết kiệm chi phí nhất. Các nhà bán lẻ như Thế giới di động, FPT Shop … có thể nhận biết được khách hàng đang thích và không thích các sản phẩm gì, từ đó có những chiến lược về nhập hàng tồn kho, chiến lược khuyến mãi và quảng cáo nhằm nâng cao doanh số và doanh thu của sản phẩm. Với những người dùng cuối, khi họ có nhu cầu về việc mua điện thoại thông minh nói riêng và các sản phẩm khác nói chung, họ có thể nhanh chóng biết được mạng xã hội và các trang tin tức đang bàn về sản phẩm nào nhiều nhất, những sản phẩm đó có ưu và nhược điểm gì từ những phân tích của hệ thống, từ đó giúp cho họ có quyết định mua hàng nhanh chóng hơn, phù hợp với nhu cầu hơn và không bị mất quá nhiều chi phí tìm kiếm thông tin.

TÀI LIỆU THAM KHẢO

[1] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. (2018). Improving language understanding by generative pre-training.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. (2017). Attention is all you need. *In NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, 6000–6010.

[3] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology, 1(1),* 4-20.

[4] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. (2017). Learned in translation: Contextualized word vectors. *In Advances in Neural Information Processing Systems*, 6294–6305.

[5] Charu, C. Aggarwal & ChengXiang, Zhai. (2012). *Mining Text Data*. Springer of Science & Business Media, New York

[6] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? *arXiv:1905.05583v2[cs.CL], Aug. 2019*

[7] Gary Miner, John Elder, IV, Andrew Fast, Thomas Hill, Robert Nisbet, and Dursun Delen. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Academic Press, Oxford

[8] Graham JW. Missing data analysis: making it work in the real world. *Annu Rev Psychol. 2009;60:549–576.*

[9] Hootsuite & We Are Social. (2019). Digital 2019: Vietnam. Retrieved from https://datareportal.com/reports/digital-2019-vietnam

[10] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *In Empirical Methods in Natural Language Processing(EMNLP)*

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. (2018). Pre-trained bert using tensorflow hub.

[12] Jacob Devlin, Ming-Wei Chang, Kenton, Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805v2[cs.CL], May. 2019*

[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. (2014). Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),* 1532–1543.

[14] Jeremy Howard and Sebastian Ruder. (2018). Universal language model fine-tuning for text classification. *In ACL,* 328–339.

[15] John Brandt. Text mining policy: Classifying forest and landscape restoration policy agenda with neural information retrieval. *arXiv:1908.02425v1[cs.CL], Aug. 2019.*

[16] Mahsa Shafaei, Niloofar Safi Samghabadi, Sudipta Kar, and Thamar Solorio. Rating for Parents: Predicting Children Suitability Rating for Movies Based on Language of the Movies. *arXiv:1908.07819v2[cs], Aug. 2019*

[17] Manish Munikar, Sushil Shakya, and Aakash Shrestha. Fine-grained Sentiment Classification using BERT. *arXiv:1910.03474v1[cs.CL], Oct. 2019*

[18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. (2018). Deep contextualized word representations. *arXiv:1802.05365v2[cs.CL], Mar. 2018*

[19] Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. (2019). Language model pre-training for hierarchical document representations. *CoRR, abs/1901.09128.*

[20] Minqing Hu & Bing Liu. (2004). Mining and summarizing customer reviews. *Association for the Advancement of Artificial Intelligence*

[21] Quoc Le and Tomas Mikolov. (2014). Distributed representations of sentences and documents. *In International Conference on Machine Learning,* 1188–1196.

[22] Sepp Hochreiter and Jurgen Schmidhuber. (1997). Long short-term memory. *Neural computation 9(8)*, 1735–1780.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. (2013). Distributed representations of words and phrases and their compositionality. *In Proceedings of the International Conference on Neural Information Processing Systems*, 3111–3119.

[24] Temax. (2019). GfK TEMAX results for Vietnam, Q3 2019. Retrieved from https://temax.gfk.com/en-vn/VND/reports/

[25 Thorsten Joachims. Transductive inference for text classification using support vector machines. (1999). *In Icml, 99*, 200–209.

[26] Wed McKinney. (2013). *Python for Data Analysis*. O'Reilly Media.

[27] Wei Yang, Haotian Zhang, and Jimmy Lin. (2019). Simple applications of BERT for ad hoc document retrieval. *arXiv:1903.10972[cs.IR], Mar. 2019*

[28] Xin Rong. word2vec Parameter Learning Explained. *arXiv:1411.2738v4 [cs.CL] 5 Jun 2016*

[29] Xuan-Son Vu, Thanh Vu, Son N. Tran, Lili Jiang. ETNLP: A Visual-Aided Systematic Approach to Select Pre-Trained Embeddings for a Downstream Task.

(2019). *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*